Simulation Standard

Connecting TCAD To Tapeout

A Journal for Circuit Simulation and SPICE Modeling Engineers

RPI VCSEL Model Released in SmartSpice

RPI VCSEL model was developed by Professor Michael Shur and his team from the Rensselaer Polytechnic Institute (RPI) [1]. A release of this model has been implemented within ***SmartSpice***, and can be accessed by setting LEVEL=4 in the diode modelcard.

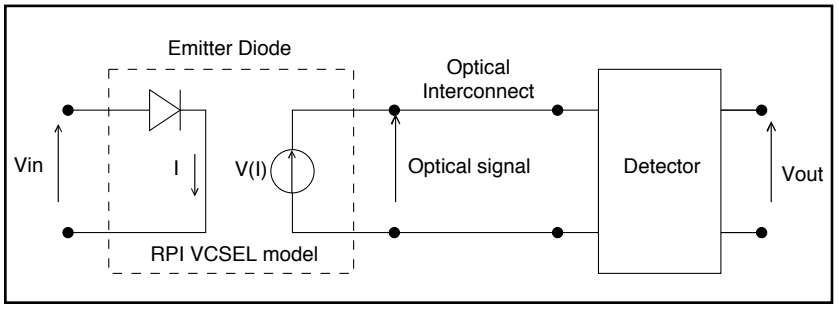

Figure 1. Optical emitter, interconnect and detector.

Mixed Photonic/Electronic Simulation

The optical devices are divided in three parts: emitters (laser diode, LEDs), detectors and interconnects. To provide a Mixed Photonic/Electronic simulation, the photonic signals are described in terms of electrical signals and can therefore be integrated in SmartSpice simulation.

RPI Vertical Cavity Surface Emitting Laser (VCSEL) model is an emitter diode whose optical output power is mapped into an electrical signal. It can therefore be connected to an optical interconnect such as a transmission line. The device is composed of electrical and optical sub-circuits. The electrical sub-circuit is a diode and the equivalent circuit of the device is a current controlled voltage source, as shown in Figure 1.

Model Description

The electric sub-circuit is based on a diode LEVEL=1 Berkeley model. The optical sub-circuit is based on the first order rate equations of semiconductor lasers, as described in [2]:

$$\frac{dN}{dt} = \frac{I}{qV} - \frac{N}{qV} - \frac{v_g \cdot A \cdot \Gamma \cdot (N-N_{tr}) \cdot S}{1 + \varepsilon S}$$

$$\frac{dS}{dt} = \gamma \cdot \frac{\Gamma \cdot N}{\tau_n} - \frac{S}{\tau_p} + \frac{v_g \cdot A \cdot \Gamma \cdot (N-N_{tr}) \cdot S}{1 + \varepsilon S}$$

These equations, involving the carrier density N and the photon density S, describe respectively the electrical properties of a semiconductor laser, and the optical behavior when some photons are produced.

The equivalent circuit of the model is shown in Figure 2.

Idiode, Ileak are diode and leakage currents, Rs is the diode resistance and Cj the junction capacitance. Il, Isp, Ig, Rp, Cp are deduced from first order rate equations (see [1]).

This model allows to simulate optical output power versus input current curves. Due to the strong thermal effects (see Figure 3.), a self-heating sub-circuit has also been added to take into account the thermal leakage current. It leads to an output-power rollover as the input current increases. It has been implemented as an internal node with thermal resistance Rth, thermal capacitance Cth, and thermal excitation Ith.

Continued on page 2 ...

INSIDE

SILVACO INTERNATIONAL

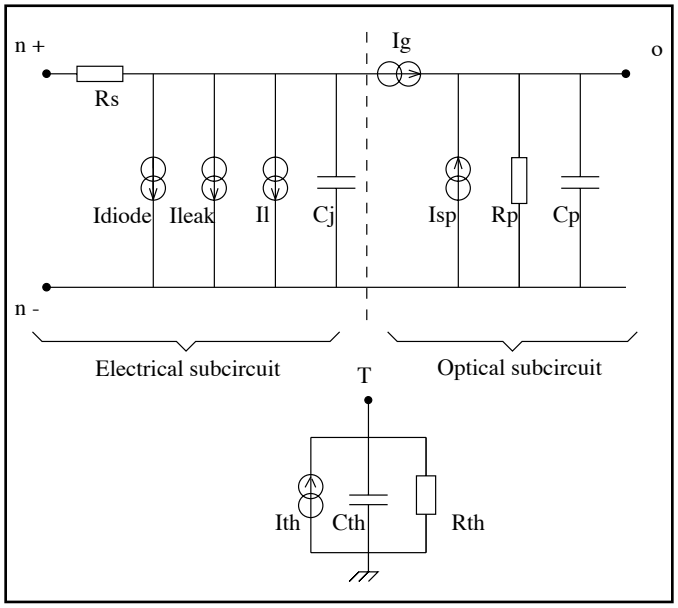

Figure 2. RPI VCSEL equivalent circuit

RPI VCSEL Model and *SmartSpice*

RPI VCSEL model has been implemented as a diode device. The third node is accessible through device declaration:

Dxxx n+ n- o mname ...

n+, n- are respectively anode and cathode of the diode, voltage bias at "o" node is the optical power output in mW. The device can be connected to optical interconnects, such as transmission lines to provide a Mixed Electronic/Photonic simulation.

The model card for RPI VCSEL includes the following parameters:

Optical Equations Parameters

Parameter	Description	Units	Default
DA	Active region depth	m	1.0E-8
TN	Electron lifetime	s	1.0E-9
NTR	Trap density	m-3	0.0
BETA	Spontaneous emission factor	-	1.0
TPH	Photon lifetime	s	1.0
GAMMA	Optical confinement factor	-	1.0
EPS	Gain compression parameter	m3	1.0
G0	Gain slope	m2	1.0
VG	Group velocity	m/s	3.0E8
LAMBDA	Wavelength of optical output	m	8.0E-6
REF	Mirror reflectivity	-	0.99
TOSS	Characteristic temperature of carrier lifetime	K	30
C0	Leakage saturation current	A/m2	1.0E-12
FRAC	Fraction of aluminum	-	0.0
T1	Fitting parameter	s	0.0
IMAX	Maximum saturation current	A	1.0
CM	Saturation knee parameter	-	1.0

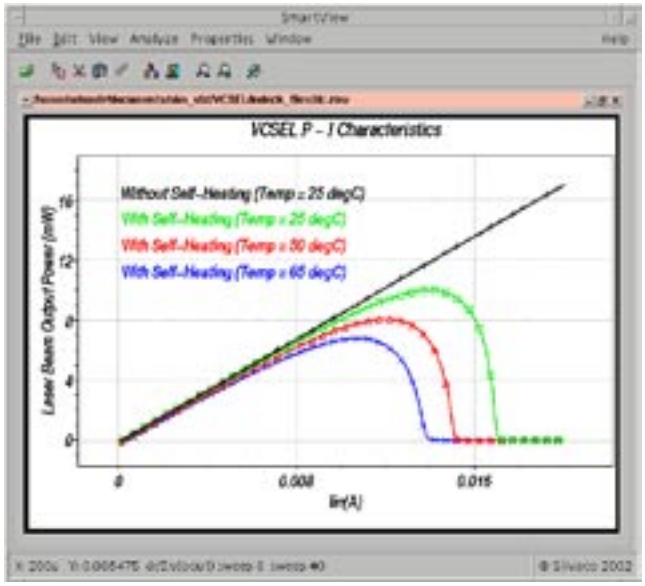

Figure 3. RPI VCSEL P - I characteristics.

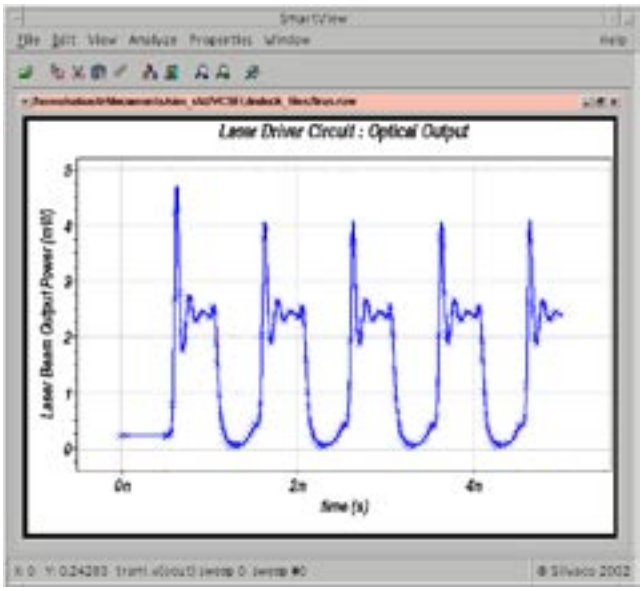

Figure 4. Optical Output of a RPI VCSEL device coupled with a Laser Driver Circuit.

Diode Parameters

Parameter	Description	Units	Default
IS	Saturation current	A/m^2	1.0E-14
TNOM	Parameter measurement temperature	K	TNOM (default)
RS	Ohmic resistance	$Ohm.m^2$	0.0
N	Emission coefficient	-	1.0
TT	Transit time	s	0.0
CJ0	Junction capacitance	F/m^2	0.0
VJ	Junction potential	V	1.0
M	Grading coefficient	-	0.5
EG	Activation energy	eV	1.43
XTI	Saturation current temperature exponent	-	3.0
FC	Forward bias junction fit parameter	-	0.5
BV	Reverse breakdown voltage	V	-5.0
IBV	Current at reverse breakdown voltage	A	1.0E-3
IKF	Forward knee current	A	0.0
ISR	Recombination current	A/m^2	0.0
NR	Recombination current slope	-	2.0
KF	Flicker noise coefficient	-	0.0
AF	Flicker noise exponent	-	1.0

Thermal Sub-Circuit

Parameter	Description	Units	Default
CTH	Thermal capacitance	$J.m^2/degC$	0.0
RTH	Thermal resistance (RTH = 0.0 : no self-heating)	$degC.m^2/W$	0.0

The model supports DC, AC, TRAN and PZ analysis. Bypass calculation and VZERO options have been implemented for a faster simulation when running large circuits. RPI VCSEL model has also been optimized to take advantage of multi-processor machines. Additional outputs have been added in ***SmartSpice*** implementation : SN and NN outputs are photon and carrier densities in the photonic device.

References

[1] J. Deng, M. S. Shur, T. A. Fjeldly, S. Baier, "CAD Tools and Optical Device Models for Mixed Electronic/Photonic VLSI", International Journal of High Speed Electronics and Systems, International Journal of High Speed Electronics and Systems, Invited, Volume 10, No 1, pp. 299-308, March 2000.

[2] R. S. Tucker, " Large-Signal Circuit Model for Simulation of Injection-Laser Modulation Dynamics", IEE Proceedings 128 (1981) 180-184.

BSIM3v3 Model Verilog-A Implementation

BSIM3v3 Model

Berkeley University BSIM3v3 model is the industry-standard, physics-based, deep-submicron MOSFET model for digital and analog circuit designs. This makes BSIM3v3 model a good candidate for implementation in Verilog-A HDL for study purpose or customized model use.

The SILVACO BSIM3v3 Verilog-A porting is based on the latest version BSIM3v3.2.4, released on December, 21st 2001. ***SmartSpice*** Verilog-A interface version 2.6.0.R has been used.

Verilog-A Porting

SILVACO BSIM3v3 Verilog-A implementation includes all the improvements of the original Berkeley model: ΔL and ΔW dependencies for different W and L devices, new capacitance model, new relaxation time model for characterizing the NQS effect, source/bulk and drain/bulk diode models. Additional model features such as Bulk-Drain and Bulk-Source GMIN, drain/ source inversion, N/P MOS type and parameter checking have also been added.

Implemented model selectors are:

```
parameter integer MOBMOD = 1; //
  Mobility model selector
parameter integer CAPMOD = 3; //
  Capacitance model selector
parameter integer NQSMOD = 0; //
  Non-quasi-static model selector
```

The latest intrinsic capacitance model (Charge Thickness Model) is available with CAPMOD=3 and is set by default. Mobility model accounts for depletion mode devices (MOBMOD=2) and body bias dependence (MOBMOD=3). The NQS model subcircuit has been added and can be turned on by setting NQSMOD to 1 (NQS model is turned off by default).

L and W dependent parameter calculation as well as parameter checking are needed only once at the beginning of the simulation. Therefore, they have been implemented using `initial_step` event for simulation time saving:

```
@(initial_step)
  begin
  ...
  nsub  = nsub  + LNSUB  * InvL + WNSUB
  * InvW + PNSUB  * InvLW;
  ngate = ngate + lngate * InvL + wngate
  * InvW + pngate * InvLW;
  ...
  if (nsub <= 0.0)
      begin
         $strobe ("Fatal: Nsub = %g is
         not positive.", nsub);
         FatalFlag = 1;
      end
  ...
end
```

DC equations are implemented in the analog part, for example Ids current:

$$Idso = \frac{\beta \cdot Vgsteff \cdot \left(1 - Abulk \cdot \frac{Vdseff}{2 \cdot (Vgsteff + 2 \cdot Vt)}\right) \cdot Vdseff}{1 + Vdseff/(Esat \cdot Leff)}$$

$$Ids = \frac{Idso}{1 + \frac{Rds \cdot Idso}{Vdseff}} \cdot \left(1 + \frac{Vds - Vdseff}{V_A}\right) \cdot \left(1 + \frac{Vds - Vdseff}{V_{ASCBE}}\right)$$

becomes in Verilog-A language:

```
analog begin
   ...
   fgche1 = Vgsteff * (1.0 - 0.5 * Abulk *
   dseff / Vgst2Vtm);
   fgche2 = 1.0 + (Vdseff / EsatL);

   gche = beta * fgche1 / fgche2;
   Idl  = gche * Vdseff / (1.0 + gche * Rds);

   Idsa = Idl  * (1.0 + (diffVds / Va));
   Ids  = Idsa * (1.0 + (diffVds / VASCBE));
   ...
   I(drain, source) <+ TYPE * Ids;
   ...
end
```

Once the Ids current is calculated, its contribution is added with <+ operator between the intrinsic drain and source nodes. Derivatives are automatically calculated so the written code is shorter than the compact model C-code and bugs in hand-written derivatives are no more possible. N/P MOS type contribution is accounted for with TYPE parameter.

The BSIM3v3 charge model is implemented the following way:

```
if (analysis("tran", "ac"))
  begin
    if (CAPMOD == 0)
      begin
        qgate  = ...;
        qdrain = ...;
      end
    else if (CAPMOD == 1)
      begin
        qgate  = ...;
        qdrain = ...;
      end
    else if (CAPMOD == 2)
      ...
  end
```

Intrinsic node charges are calculated according to `CAPMOD` model parameter. `analysis("tran", "ac")` function call can be used to compute charges only in transient and ac analysis. Then, extrinsic elements such as overlap capacitances charges are calculated and added to the corresponding node, for example:

```
qgdo = Cgdo * Vgd;
Qgate  = qgate  + qgdo; // qgdo is
  gate/drain overlap capacitance
  charge
Qdrain = qdrain - qgdo;
```

The resulting charge current contribution is added to the circuit by using ddt time derivative operator:

```
I(gate)  <+ TYPE * ddt(Qgate); // Qgate
  is the total node charge
I(drain) <+ TYPE * ddt(Qdrain);
```

Validation

In transient analysis, a one-shot trigger test circuit supplied by Berkeley University team has been successfully run with BSIM3v3 Verilog-A model, with a reasonable simulation time. In AC analysis as well, an operational amplifier has been successfully simulated.

To improve the model convergence especially with large circuits, the MAXDELTA nature attribute is used to systematically limit the per-iteration voltage bias changes. GMIN conductances have been added to drain/bulk and source/bulk branches to improve convergence. GMIN is a model parameter and set to 1e-12 Ω^{-1} by default.

For Verilog-A implementation validation, results have been compared with ***SmartSpice*** internal BSIM3v3 model level=8.

Conclusion

Berkeley BSIM3v3.2.4 model has been successfully implemented in Verilog-A at SILVACO. The fact that the derivatives are automatically calculated reduces dramatically the implementation time and avoid introducing bugs in derivative calculation. It also increases the readability, shortening the 20,000 lines original Berkeley C-code to a 3,000 lines Verilog-A module. The Verilog-A simulation fit the results with ***SmartSpice*** internal model and shows a good convergence on large circuits with the help of voltage limitation. The Verilog-A model is freely available on SILVACO website **(http://www.silvaco.com).**

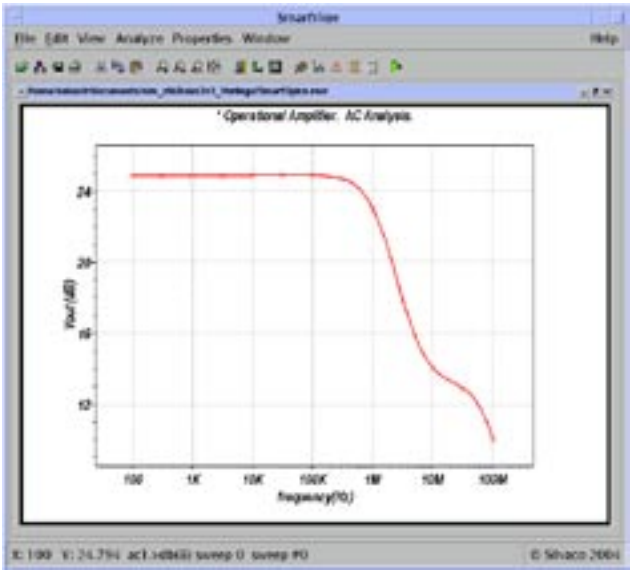

Figure 1. Operational amplifier AC output with BSIM3v3 Verilog-A model

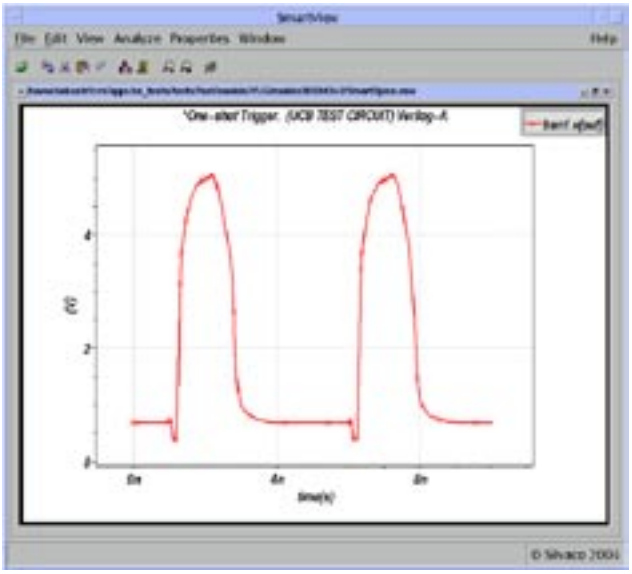

Figure 2. One-shot trigger output with BSIM3v3 Verilog-A model

New Device Model Card Approach

Introduction

Normally a single active device model is extracted to cover a range of device geometries and temperature. Sometimes this single scaleable model is not sufficient to describe all the changes in output characteristics over the range of geometry and temperature required. The total range of geometry and temperature is then broken up into regions and a model produced for each of these sub-sets of devices. This is the basis of a binned model and can lead to discontinuities at the bin boundaries as the model card is changed. To get round this problem ***SmartSpice*** has introduced a new functionality to allow the user to go back towards a single scaleable model card via the use of a function rather than a single parameter value. This new powerful algorithm allows the user to specify a formula linking in other device model card parameters as so giving a continous multi-dimensional function. This then gets round the problems of binning and gives the potential of a more accurate model fit to the device output characteristics.

Setting up the Model's Parameter

SmartSpice offers new algorithm for setting up a model for SOI, TFT, MOSFET, JFET, MESFET, bipolar and diode. This new feature is intended to substitute for the binning scheme and gives the user a powerfull mechanism for device modeling. The new algorithm allows the user to specify a formula for any model's parameter. The formula describes the model across a wide range of data and target model parameters and is represented by a continuos multi-dimensional function.

Syntax:

```
m0 B0_x10 adr7 B0_x11 vss nenh L=5.1e-06 W=2.11e-06
.model nenh NMOS
 ...
+VERSION=3.1 +Vth0='0.9302 + 0.125E-012/(W*L) +0.001*L-0.012*W'
...
```

In the example above the model card parameter Vth0 is driven by width and length which comes from a set of instance parameters. For device m0 ***SmartSpice*** will use a model with parameter Vth0 calculated using L=5.1e-06 and W=2.11e-06.

Building up the Model

An abstract model nenh is used to generate either a table of Vth0 values or create a copy of absract model values. ***SmartSpice*** uses both algorithms. Table generation algorithm is used by default and is recommended. There is a variable "modelalg" which must be specified in the initialization file(.SmartSpice.in(Unix) and smspice.set(Windows)) to switch the internal algorithm. If modelalg=0(default) then ***SmartSpice*** builds up the table with the parameter's values. If modelalg=1 then ***SmartSpice*** creates a separate model for each instance.

Perfomance in the New Model Approach

modelalg=0 is a memory conserving approach and has a higher speed during the parser phase. ***SmartSpice*** bypasses the regular model creation procedures and builds up an optimized (quick selection) table. During the simulation phase ***SmartSpice*** operates with the table to set up a correct model parameter value. The overhead for modelalg=0 is near 3%-6% of total simulation time. The post-processing phase takes less time. Use ***SmartSpice*** shell command "cmcstat" to get the absolute overhead during the simulation.

modelalg=1 is a more memory consuming approach. If the netlist contains a lot of different geometries (a lot of groups) ***SmartSpice*** will allocate memory for each group. The parser phase can take 4-5 times longer in comparision with the modelalg=0.

During simulation modelalg=1 is more preferable then modelalg=0 because ***SmartSpice*** does not need to make a selection of the model's parameter from table. Post processing takes more time.

SmartSpice shell Command for Checking up the Model's Parameter table

All devices are grouped with respect to the arguments of the formula used for the model parameter. To see how many different groups are created and the model parameter target value ***SmartSpice*** offers the shell command "cmcstat". The output after the use of "cmcstat" is shown below with modelalg=0:

```
--> cmcstat

****internal information about CMC format model use

****

Total Overhead During Simulation = 0.000000

Model nenh

vth0=0.9302 + 0.125e-012/(w*l) +0.001*l-0.012*w

group 0 W=1.050000e-05 L=1.440000e-06 9.384671e-01
```

group 1 W=1.050000e-05 L=1.450000e-06 9.384101e-01

group 2 W=1.050000e-05 L=1.460000e-06 9.383538e-01

group 3 W=2.100000e-06 L=1.100000e-06 9.843125e-01

group 4 W=2.110000e-06 L=2.100000e-06 9.584103e-01

group 5 W=2.120000e-06 L=1.100000e-06 9.838020e-01

group 6 W=2.130000e-06 L=1.100000e-06 9.835504e-01

group 7 W=2.140000e-06 L=1.100000e-06 9.833011e-01

group 8 W=2.150000e-06 L=1.100000e-06 9.830541e-01

group 9 W=2.160000e-06 L=1.100000e-06 9.828094e-01

Output after the use of "cmcstat" is shown below when modelalg=1:

--> cmcstat

****internal information about CMC format model use****

[holder model]=nenh

group 0 W=1.050000e-05 L=1.440000e-06

group 1 W=1.050000e-05 L=1.450000e-06

group 2 W=1.050000e-05 L=1.460000e-06

group 3 W=2.100000e-06 L=1.100000e-06

group 4 W=2.110000e-06 L=2.100000e-06

group 5 W=2.120000e-06 L=1.100000e-06

group 6 W=2.130000e-06 L=1.100000e-06

group 7 W=2.140000e-06 L=1.100000e-06

group 8 W=2.150000e-06 L=1.100000e-06

group 9 W=2.160000e-06 L=1.100000e-06

User's Oriented Errors and Warning Messages

Conflict of Names in the .PARAM and Formula Arguments

New model approach algorithm reports about errors. In the case of using the same names for parameter in .param statement and arguments in formula model's parameter (Berckely approach) ***SmartSpice*** will issue the warning.

Example:

.param w=1

.model nenh NMOS

+Vth0='0.9302 + 0.125E-012/(w*l) +0.001*l-0.012*w'

Warning:

New model format in use : member [w] in the expression [0.9302 + 0.125e-012/(w*l) +0.001*l-0.012*w] of model [nenh](file ./bsim3v3.mod

/home/user/MYINDECK/CMC/bsim3v3.mod) is treated as device geometry and overrides the value of parameter [w] in the .PARAM

User must check either formula in the .model card or parameter in the .PARAM.

Arguments in the Formula for Model Card Parameter do not Correspond to the Set of Device's Parameters

Example:

q1 cq bq gnd1 gnd QNLREF m=1 areac =45

.modelqnlrefnpn (

...

+ BF='0.5+area+1'

Fatal Error: CMC model: Model qnlref can not find instance parameter 'area' in the specified device's line

q1 cq bq gnd1 gnd qnlref m=1 areac =45

used in the expression '0.5+area+1'

Error on lines:

29 : .modelqnlrefnpn (

30 : *+BF=100 31 : +BF='0.5+area+1'

...

cannot evaluate expression '0.5+area+1'

SmartSpice's System Internal Messages

SmartSpice prints the error message "Device type "CAP32" is not supported in the enchanced modeling algorithm" if the parse finds inconsistencies in the parser phase. In this case user must check the .modelcard for device CAP32.

System message "CMC model: module found fatal error, parameter table and corresponding value table are not syncronized" is issued by ***SmartSpice*** if internal engine fails to find correspondences in the internal tables. User must report that message to the ***SmartSpice*** support.

Multithreading Support in the New Model Approach

modelalg=0 and modelalg=1 support multithreading in ***SmartSpice.*** The user does not need to specify any extra input for ***SmartSpice***, only -P n, where N-is a number of CPUs.

Enchancements of the New Modeling Approach

The set of allowed instance's parameters which can be used in the formula for a model card parameter is the subject for increase later. The core of the algorithm is scalable.

Typegroup	Technology	Internal name	Info	Level	the list of supported parameters
npn pnp lpnp	BJT	BJT	Bipolar Junction Transistor	1, 2	area, areb, areac
		VBIC	VBIC Bipolar Junction Transistor	5	area, areab, areac
		HICUM	HICUM Bipolar Junction Transistor	6	area, areab, areac
		PBJT	Mextram BJT (Philips)	503, 504	area, areab, areac
		MODELLA	Philips TPL500 Bipolar Transistor	500	
		HBT	Hetero-Junction Bipolar Transistor	20	area, areab, areac
nmos pmos ntft ptft	SOI	BSIM31SOI	Berkeley SOI MOSFET model version 1 (level 25)	25	w, l
		BSIM3SOI2DD	Berkeley SOI MOSFET model version 2 (level 27)	27	w, l
		BSIM3SOI2FD	Berkeley SOI MOSFET model version 2 (level 26)	26	w, l
		BSIM3SOI2PD	Berkeley SOI MOSFET model version 2 (level 29)	29	w, l
		BSIM3SOI3	Berkeley SOI MOSFET model version 3 (level 33)	33	w, l
		UFS	University of Florida SOI Model (level 21)	21	w, l
		LETISOI	CEA/LETI SOI MOSFET model	32	w, l
	TFT	TFT	MOS field-effect transistor	15	w, l
		PTFT	PolySi TFT model	16	w, l, area
		MOS15	MOS15 TFT Model	35	w, l
		MOS16	RPI Poly-Si TFT Model	36	w, l
	MOSFET	MOS123	MOS field-effect transistor	1, 2, 3	w, l
		BSIM1	Berkeley Short Channel IGFET Model	4, 13	w, l
		BSIM3	Berkeley Short Channel IGFET Model) Version-3 (level 81	81	w, l
		BSIM3v3	Berkeley Short Channel IGFET Model Version-3 (level 8, 49, 53)	8, 49, 53	w, l
		BSIM3M	Modified Berkeley Short Channel IGFET Model Version 3 (level 7,10,47)	7, 10, 47	w, l
		BSIM4S	Berkeley Short Channel IGFET Model-4 (level 14, 54)	14, 54	w, l
		MOS11	Philips MOS11 model	11,63	w, l
		MOS31	MOS31 MOSFET Model	30, 31, 40	w, l
		MOS20	Philips MOS20 LDMOS model	20	w, l
		EKV	EKV MOSFET Model	44	w, l
		BSIM3H	High-Voltage MOSFET Model (level 88)	88	w, l
		HISIM	Hiroshima University STARC IGFET Model (level 111)	111	w, l
njf pjf nmf pmf	JFET/ MESFET	JFET	Junction/Schottky contact field-effect transistor	1, 2, 3, 4, 5, 6	w, l, area
d	Diode	DIO	Junction Diode	1, 3	w, l, area
		DIO2	Fowler-Nordheim Diode	2	w, l , area
		DIO500	Diode Level 500	500	
		JCAP	Junction Capacitor	9	area
		LAS1	VCSEL model	4	area
c cap	Capacitance	FCAP	Ramtron Ferroelectric Capacitance Model	5	
		FRMC	Ramtron Ferroelectric Capacitance Model	6	

Table 1. Correspondence between different types of devices and device's parameters which can be used in the model's parameter.

Calendar of Events

April

1	GOMAC Tech - Tampa, FL
2	GOMAC Tech - Tampa, FL
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	ISPSD - Cambridge, UK
15	ISPSD - Cambridge, UK
16	ISPSD - Cambridge, UK
17	ISPSD - Cambridge, UK
18	
19	
20	
21	
22	
23	P2ID - Paris, France
24	P2ID - Paris, France
25	P2ID - Paris, France
26	
27	Ultra Shallow Junction - Santa Cruz, CA
28	Ultra Shallow Junction - Santa Cruz, CA
29	Ultra Shallow Junction - Santa Cruz, CA
30	Ultra Shallow Junction - Santa Cruz, CA

May

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	IPRM - Santa Barbara,CA
13	IPRM - Santa Barbara,CA
14	IPRM - Santa Barbara,CA
15	IPRM - Santa Barbara,CA
16	IPRM - Santa Barbara,CA
17	
18	
19	GaAs MANTECH - Scottsdale, AZ
20	GaAs MANTECH - Scottsdale, AZ
21	GaAs MANTECH - Scottsdale, AZ
22	GaAs MANTECH - Scottsdale, AZ
23	
24	
25	
26	
27	
28	
29	
30	
31	

Bulletin Board

See Silvaco at GOMAC

GOMAC is an ITAR-restricted conference established primarily to review developments in micro-circuit applications for government systems. Established in 1968, the conference has focused on advances in systems being developed by the Department of Defense and other government agencies and has been used to announce major government microelectronics initiatives such as VHSIC and MMIC, and provides a forum for government reviews.

See Silvaco at GaAs MANTECH

To stay ahead of the aggressive scaling of the silicon technology, compound semiconductor technologies continue to scale down to the nano-scale dimension. In addition, the wide-bandgap and narrow-bandgap materials are making rapid advances and may move to the manufacturing arena in the near future.

If you would like more information or to register for one of our our workshops, please check our web site at ***http://www.silvaco.com***

Hints, Tips and Solutions

Colin Shaw, Applications and Support Engineer

Q. How do I make my long simulation more memory efficient ?

When you run a long simulation time analysis the evaluated time points are normally all held in memory until the end time is reached. All the data is then written out to the output rawfile. This means a large amount of system memory can be used up and also has to be tracked. If in the input deck the line ".OPTIONS RAWPTS=300 POST" is included then as soon as the maximum number of points is reached given by "RAWPTS=300" then all the vector data is saved to the raw data output file and the memory is free to be re-used. In this way the output rawfile is incrementally increased in size every time this limit is reached. The memory required by the simulation run is therefore reduced, and with less memory to manage, the simulation is run faster. This is particularly useful on the PC platform where some of the memory is required for the operating system and RAM size is less than 1Gig.

Q. What is the difference between .MODIF & .ALTER for simulation run variance ?

The .ALTER is used to globally change the circuit topology like slotting in another sub-circuit or changing a component. This change in the circuit is like submitting a new deck and can be incorporated in a distributed system among a group of computers to get maximum utilization of the resources. The .MODIF is more for changing parameter values associated with a model etc.

Q. My simulation fails with no convergence what extra information can I obtain ?

The spice simulation of a circuit is based on a good model of the elements and sensible circuit construction. If you include the line ".OPTIONS EXPERT" you will get a lot of reporting turned on showing model card parameter evaluation and floating nodes of the circuit. It would be sensible to cut down the simulation time before turning this option on to save too much duplication of the errors everytime they are encountered. This reporting gives the user more information to construct a better simulation deck. Commonly the model construction can be at fault and this option allows the user to go back to his model provider with highlighted problem areas. The other common problem is having an isolated section of circuitry without a DC path to ground. This can happen if a section of circuitry is connected only through a set of capacitance elements for example. Here the user will need to add conductance to ground to these nodes to allow spice to achieve a DC solution .OPTIONS DEPATH =<val> .The spiceusr1.pdf in chapter 2.12 contains options broken up into categories like convergence and we welcome any customer feedback on the manual information.

Q. I kill my SmartSpice program but the license is not available in Linux ?

When you invoke ***SmartSpice*** you can run it in the Foreground or the background by use of the "&" character at the end of the command line. Programs run in the background return the prompt straight away to the invoking window, programs run in the foreground wait for the completion of the program to return to the prompt. If the program is run in the foreground and then <cntrl>"C" is done the operating system still holds onto the process hence License is not available. To put a process into the background the <cntrl>"Z" should be done AND then the command "bg" to place in the background and the system will release the process. The License is therefore available for the next version of ***SmartSpice.*** (There is a License time-out feature but this will take a while to release the License)

Q. I know SPICE is an Analog simulator but can I generate a digital source ?

There are 2 steps to generating a digital source, first the waveform timing and second the pulse train of "0"'s and "1".

A SPICE input deck example is:

Digital source config.

```
* PWLFILEDESC.
.param tri=150ps tfi=150ps
* PWLFILEDESC format is:
*  ( datatype start delay  vlow vhigh  fall
rise )
* datatype= 1 - time current pairs, = 2
time voltage pairs
*
vin inp 0  pwlfiledesc( 2  0  3ns  0v  3.3v
'tri' 'tfi' ) pwlfile test3.dat


vcc 0   vss    dc -3.3v
m1 2  inp  vss  vss  pm  w=32u l=1.6u
m2 2  inp  0    0    nm  w=10u l=1u
C1 2 0 0.01f
```

```
.MODEL  NM    NMOS   LEVEL    = 49
.MODEL  PM    PMOS   LEVEL    = 49

.tran 0.1ns 50ns 0.1ns
.print v(inp) v(2)
.temp 125
.option nomod
.END
```

and the referenced file (test3.dat) contains the digital string "010110010"

What noise analysis can I do on my circuit in *SmartSpice*?

There are 3 ways to take account of noise in your circuit. First there is the traditional .NOISE analysis used to calculate the noise of each component. From there the spectral density is obtained and the integrated noise over the specified frequency range used to calculate the output and the input referred noise. Second there is the .TRAN NOISE which calculates the noise contributions of each circuit node and solves the stochicometric set of equations for each time point in the transient analysis. And finally there is a new spice element "N" introduced in ***SmartSpice*** 2.3.8.C, and higher, to allow users to inject noise at a particular circuit node. This noise is in addition to the element sources of noise and allows for contributions like substrate referred noise from other parts of the circuit and general circuit succeptability to external noise sources.

Q. My long simulation run consumes all resources and then freezes ?

This happens when you use a PC system with less than 1Gig of RAM or a small amount available hard disc space and a very long simulation time recoding all circuit node activities. A new feature ".OPTIONS SAFEMODE" was introduced to warn of this happening and it stops the simulation when either the hard disc space or the available memory gets below 50Megs. It is then up to the user to close other programs and free more resources or continue at risk running out of space. The other ways are to limit the simulation time to something more reasonable or limit the vectors that are saved rather than having everything saved even circuit nodes that are not critical. To do this instead of the ".SAVE ALL" you can use 2 pre-defined macros to record vectors from part of the whole circuit.

Xx - defines

Q. My simulation fails because SmartSpice cannot find a binned model, what does this mean ?

Normally when a MOS model is extracted for a range of devices it is scaleable over the range of device geometries i.e. it is a continuous varying function over the required

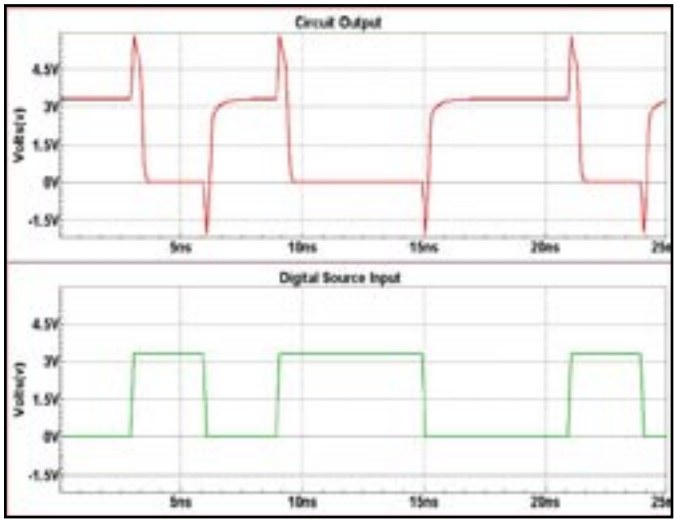

Figure 1. Digital source - input and circuit output.

operating region for all device geometries. Sometimes there is too much variation in the output characteristics to be covered by one continuous model parameter set e.g. "straight" and "dog-bone" layout designs which contain very different electric field patterns. The total operating region is therefore broken up into sub-sections and a model produced for each of these subsets of device geometry. This is a binned model where each region is a different set of model card values for a smaller range of device properties like width, length and temperature. In this way a group of model card parameter sets can be used to cover a wide variation in say gate width and length variations not possible from a single scaleable model. In the simple case these bins are ranges of Width and Length transistor geometries that say what model card parameter set should be used. The only problem with this approach is a discontinuity at the boundary of one model set to another and can be thought of as trying to approximate a curve with a set of straight lines. Your error is because the device geometry is not covered by any of the specified ranges in the model library. Typically the binned model will have a model name of say nch.1, nch.2, nch.3 etc. and you device geometry is not allowed for in the say Lmin to Lmax range of any of these binned model sections.

Call for Questions

If you have hints, tips, solutions or questions to contribute, please contact our Applications and Support Department

Phone: (408) 567-1000 Fax: (408) 496-6080

e-mail: support@silvaco.com

Hints, Tips and Solutions Archive

Check our our Web Page to see more details of this example plus an archive of previous Hints, Tips, and Solutions

www.silvaco.com

Your Investment is Safe

20 Years and Growing

Financially Rock-Solid

Fiercely Independent

Analog/MS EDA Design Leader

We are NOT For Sale

SILVACO
INTERNATIONAL

USA Headquarters:

Silvaco International
4701 Patrick Henry Drive, Bldg. 2
Santa Clara, CA 95054 USA

Phone: 408-567-1000
Fax: 408-496-6080

sales@silvaco.com
www.silvaco.com

Contacts:

Silvaco Japan
jpsales@silvaco.com

Silvaco Korea
krsales@silvaco.com

Silvaco Taiwan
twsales@silvaco.com

Silvaco Singapore
sgsales@silvaco.com

Silvaco UK
uksales@silvaco.com

Silvaco France
frsales@silvaco.com

Silvaco Germany
desales@silvaco.com

*Products Licensed through Silvaco or e*ECAD*