Authors: Navid Paydavosi (navidp@eecs.berkeley.edu)

❖ *Bug fixes*

The redundancy in the calculation of RSourceGeo and RDrainGeo was removed:

// Component 1: Extension Resistance
- T0 = max(0, LSP + deltaL - LDG * log(NSD / NSDE));
- T1 = T0 + LDG / ln(10) * (NSD / NSDE - 1.0);
- Rext = rhoext / (HFIN * TFIN * NFIN) * T1;

The parameter check warnings for **PHIBE, PTWG, TNOM, PDIBL2,** and **NTNOI** were modified:

- if(PHIBE_i < 0.2 && BULKMOD != 0) begin
  $strobe("Warning: PHIBE_i = %e is less than 0.2, setting it to 0.2.", PHIBE_i);
  PHIBE_i = 0.2;
  end

- if (PTWG_i < 0) begin
  $strobe("Warning: PTWG_i = %e is negative, setting it to 0.", PTWG_i);
  PTWG_i = 0;
  end

- if(TNOM < -`P_CELSIUS0) begin
  $strobe("Warning: (TNOM=%e) < -`P_CELSIUS0. Set to 27 C.", TNOM);
  Tnom = `REFTEMP; //`REFTEMP is in Kelvin i.e. 300.15 K
  end else begin
  Tnom = TNOM + `CONSTCtoK;
  end

- if(PDIBL2_i < 0) begin
  $strobe("Fatal: PDIBL2_i = %e is negative.", PDIBL2_i);
  end

- if(NTNOI < 0) begin
  $strobe("Warning: NTNOI = %e is negative. Set to zero.", NTNOI);
  NTNOI_i = 0;
  end else begin
  NTNOI_i = NTNOI;
  end

Typo in **LAIGD1**,**NAIGD1**, and **PAIGD1** definitions were fixed:
+ parameter real LAIGD1 = LAIGS1;
+ parameter real NAIGD1 = NAIGS1;
+ parameter real PAIGD1 = PAIGS1;

The VTH definition equation was modified:

- VTH = VFB + Vtm * lln(T2/T3) + dvch_qm + phib + qbs + Vtm + dvth_all - DELVTRAND;
+ VTH = VFB + devsign*(Vtm * lln(T2/T3) + dvch_qm + phib + qbs + Vtm + dvth_all - DELVTRAND);

The reported issues on operating-point information were addressed:

```
- QD = devsign * qd - qgd_parasitic - (CGEOMOD == 1 ? qgd_fr : 0) - devsign * Qed;
+ QD = devsign * qd - qgd_parasitic - (CGEOMOD == 1 ? qgd_fr : 0) - devsign * Qed + qds_fr;

- QS = devsign * qs - qgs_parasitic - (CGEOMOD == 1 ? qgs_fr : 0) - devsign * Qes;
+ QS = devsign * qs - qgs_parasitic - (CGEOMOD == 1 ? qgs_fr : 0) - devsign * Qes - qds_fr;

- CGBOV = - devsign * ddx(- devsign * Qeg, V(e));
+ CGBOV = - devsign * ddx(Qeg, V(e));
```

❖ *Enhancements*

The following warning for **PCLM** was added:

```
+ if(PCLM_i < 0) begin
+         $strobe("Warning: PCLM_i = %e is negative.", PCLM_i);
+ end
```

Long channel DIBL (also called Drain-Induced Vth Shift or DITS) was added. New Parameters **DVTP0** and **DVTP1** were introduced:

```
+ parameter real DVTP0    =  0;                  // Coefficient for Drain-Induced Vth Shift (DITS)
+ parameter real DVTP1    =  0;                  // DITS exponent coefficient

+ dvth_dibl   =  -ETA0_a * Theta_DIBL * vdsx + DVTP0 * pow(vdsx, DVTP1);
```

Average charge weighing factor was added through the new parameter **CHARGEWF:**

```
+ parameter real CHARGEWF =  0 from [-1:1];          // Average Channel Charge Weighting Factor,
                                                          +1:source-side, 0:middle, -1:drain-side
+ if(CHARGEWF != 0)
+         qia2 = 0.5 * (qis + qid) + CHARGEWF * (1.0-lexp(-T0))* 0.5 * dqi;
+ else
+         qia2 = 0.5 * (qis + qid);
…

// *** Mobility Degradation ***
+ Eeffm =  EeffFactor * (qba + eta_mu * qia2);    // in the unit of MV/cm
+ T2    =  pow(0.5 * (1.0 + abs((qia2) / qb0)), UCS_t);
…

// *** Mobility Degradation for C-V ***
+ Eeffm_cv =  EeffFactor * (qba + eta_mu_cv * qia2);    // in the unit of MV/cm
```

Length scaling of SCE and SS were decoupled. The new binnable parameter **DVT1SS** was introduced:

```
+ parameter real DVT1SS   =  DVT1;                  // Subthreshold Swing exponent coefficient

+ tmp = DVT1SS_i * Leff / scl + 1.0e-6;
+ if (tmp < 40.0)
+         Theta_SW    =  0.5 / (cosh(tmp) - 1.0);
```

```
+ else
+       Theta_SW    = exp(-tmp);

- cdsc      =  Theta_SCE * (CDSC_i + CDSCD_a * vdsx);
+ cdsc      =  Theta_SW * (CDSC_i + CDSCD_a * vdsx);
```

A linear length scaling equation for **PHIG** was added:

```
+ parameter real PHIGL   =  0;                  // Length dependence of Gate workfunction, eV/m
+ PHIG_i = PHIG_i + PHIGL * Leff;
```

New **NFIN** scaling equations and parameters were added for the following parameters:

- PHIG
- CDSC
- CDSCD
- CDSCDR
- NBODY
- VSAT
- VSAT1
- VSAT1R
- U0

The scaling equations have the following general format:

$$PARAM_N = PARAM \times \left[1.0 + \frac{PARAMN1}{NFIN} \times ln\left(1.0 + \frac{NFIN}{PARAMN2}\right)\right]$$

Temperature dependence on **ETA0** and **ETA0R** was added.

```
+ parameter real TETA0    =  0.0;              // Temperature dependence of DIBL coefficient, 1/K
+ parameter real TETA0R   =  TETA0;        // Temperature dependence of Reverse-mode DIBL
                                                  coefficient, 1/K
+ ETA0_t   =  ETA0_i  * (1.0 + hypmax( TETA0  *delTemp, -0.9, 1e-4));
+ ETA0R_t  =  ETA0R_i * (1.0 + hypmax( TETA0R *delTemp, -0.9, 1e-4));
```

The default values of **COVS** and **COVD** were at the unrealistic value of 25 pF. The default values were set to zero.

For CGEOMOD=1 only, **GEO1SW**=1 now enables the parameters **COVS, COVD**, **CGSP**, and **CGDP** to be in F *per* fin *per* gate-finger *per* unit channel width. The default value of **GEO1SW** switch is zero.

To be consistent with SOI, **RTH** and **CTH** equations in the self-heating sub-model were updates as follows:

```
+   gth = NF * (WTH0 + NFIN * FPITCH) / RTH0;
+   cth = CTH0 * NF* (WTH0 + NFIN * FPITCH);
```

With these equations, **WTH0** is the parameter for thermal resistance spreading per gate finger, consistent with the **WTH0** definition in BSIM-SOI.

The initial guess for the surface potential calculation was improved. The model is now infinitely scalable with respect to TFIN and NBODY without any clamping on NBODY.

Old Code:

```
F0 =  (vgsfbeff - phipert - vch)/T14 - F1;
// Initial guess for sub-threshold
z1 =  atan(lexp(F0-T11));
// Initial guess for strong inversion
if (F0 > `EXPL_THRESHOLD) T0 =  F0;
else T0  =  lln(1.0 + lexp(F0));
z2 =  atan(2*T0*Inv_r1pi / T12);
// initial guess
g0 =  min(z1, max(z2, 1e-15));
if (g0 > g0max || g0 < g0min) begin
    if(g0 > g0max) g0  =  g0max;
    else g0  =  g0min;
end
```

New Code:

```
//Juan and Navid's Initial Guess 3/2013----------------------
F0  = (vgsfbeff - phibulk - vch)/T14 - F1;
T0  =  phibulk*aab;
T1  =  phibulk/T14;
z1  =  (2.0*r1-1.0) * T1 + r2 * T0;
guessA = 2.0 * r1 * T1;
guessA = guessA/(1.0-exp(-guessA));
 z2  = z1 + ln((sqrt(guessA)/r1));
if((F0-z2)>-0.3/T14) begin
            if ((F0-z2)< 20)
                        T2 = (0.5/guessA)* ln(1.0 + exp(2.0*(F0-z2))) + 1.0;
            else
                        T2 = (0.5/guessA)*(2.0*(F0-z2)) + 1.0;
            T3 = pow(T2, 2.0);
            guessB = (guessA/r1) * sqrt(T3-1.0) * exp(-T1);
end else
            guessB =  exp(F0-z1-T1);
guessB = 1.0/((1.0/ guessB)+(1.0/(0.5*`M_PI)));
g0=guessB;
```

The new code shows a smooth behavior for the surface potential and returns non-negative charge at S/D ends for high values of NBODY.

Unused model parameters NINTNOI and PINTNOI were removed.

Minor updates in the technical manual including equations 3.332 and 3.334; the equations were updated in manual to be consistent with the code.