UC Berkeley, BSIM Group
Authors: Navid Paydavosi (navidp@eecs.berkeley.edu), Sriramkumar V. (siram@eecs.berkeley.edu)

Bug_fixes

- **The definitions of the physical intrinsic capacitances, CEDI and CEEI, were corrected:**
  - CEDI = +ddx(QBI, V(di));
  + CEDI = -ddx(QBI, V(di));

  - CEEI = -ddx(QBI, V(e));
  + CEEI = +ddx(QBI, V(e));

- **Gate Resistance Model**
  Only the component of the gate contact that is over the channel should be multiplied by 1/3 or 1/12:
  - Rgeltd = (RGEXT + RGFIN * NFIN) / ((NGCON == 2 ? 12.0 : 3.0) * NF);
  + Rgeltd = (RGEXT/NGCON + (RGFIN * NFIN) / (NGCON == 2 ? 12.0 : 3.0)) / NF;

- **Issues/bugs related to NQSMOD=3**
  1. `IntrinsicSource and `IntrinsicDrain were directly used without a definition in bsimcmg_load_nqsmod3_segments.include. `IntrinsicSource and `IntrinsicDrain *were* replaced by “si” and “di”, respectively.
  2. Since no NQSMOD=3 model is invoked when GEOMOD !=3 and COREMOD != 0, all the calculations related to NQSMOD=3 should take effective with model flags GEOMOD and COREMOD together. Modifications were made to IIMOD=1, IIMOD=2, noise, and *Operating-Point information* calculations.
  3. The drain current in NQSMOD=3 is represented by “idsN” not “ids.” Modifications were made to the self-heating mode, power calculations (just before *Operating-Point information* calculations around line #3666) to fix it.
  4. NQSMOD=3 was disabled to perform an offline study.
     - parameter integer NQSMOD = 0 from [0:3];
     + parameter integer NQSMOD = 0 from [0:2];

- **Geometry Dependent Source/Drain Resistance**
  - parameter real LDG = 5.0e-9 from [0:inf];
  + parameter real LDG = 5.0e-9 from [0:inf);

- **Preventing Nan issue in IIMOD=1**
  - T1 = - BETA0_t / diffVds;
  + T1 = - BETA0_t / (diffVds + 1.0e-30);

- **Effective Channel Length**
  - deltaL = LINT + LL * lexp(-Lg / LLN);
  + deltaL = LINT + LL * pow(Lg, -LLN);

Improvements:

- **Parameter NFIN has been converted from integer to real to enable optimization.**
  Along the same lines variable declaration NFINtotal is now real instead of an integer.

- **Coding Style**
  For _BINNABLE_ parameters, the bounds are now applied to the "Effective PARAM" (PARAM_i) instead of the root term (PARAM). These include *K0SI*, *DSUB*, *DVT1*, *PSAT*, and *PHIBE*.

- **Parameters *DELTAVSAT*, *DVTSHIFT*, and *KT1* are now binnable.**

- ***Vfbsd* for Poly and Metal Gates**

  The definition in BSIM-CMG106.0.0:

  vfbsd = devsign * (PHIG_i - (EASUB + 0.5*Eg - devsign * (0.5*Eg - hypsmooth(0.5*Eg –Vtm * lln(NSD/ni), 1.0E-4))));

was applicable to metal gates (i.e., NGATE=0) *only*. A separate expression for NGATE>0 (Poly) was encoded into BSIM-CMG106.1.0:

vfbsd = devsign * ((~~EASUB~~ + hypsmooth(0.5*Eg - Vtm * lln(NGATE/ni), 1.0E-4))- (~~EASUB~~ + 0.5*Eg - devsign * (0.5*Eg - hypsmooth(0.5*Eg – Vtm * lln(NSD/ni), 1.0E-4))));

- **Quantum Mechanical Effects**

  a) The new parameter *QMTCENCVA* (binnable) was introduced to replace *QMTCENCV* for the accumulation region capacitance.

  b) Some bugs were fixed for variable "Tcen" calculations in both inversion and accumulation region.

- **Removed nVtm in the poly-depletion correction term**

In BSIM-CMG106.0.0

```
- if(NGATE_i>0) begin
-         if(GEOMOD != 3) r2 =  4.0 * nVtm * epssub / (`q * TFIN * TFIN * NGATE_i);
-         else       r2 =  2.0 * cpoly * r1 * r1 * nVtm;
- end else begin
-         r2 =  0.0;
- end
```

In BSIM-CMG106.1.0

```
+ if(NGATE_i>0) begin
+         if(GEOMOD != 3) r2 =  4.0 * Vtm * epssub / (`q * TFIN * TFIN * NGATE_i);
+         else          r2 =  2.0 * cpoly * r1 * r1 * Vtm;
+ end else begin
+         r2 =  0.0;
+ end
```

- **Coulomb Scattering**

Using the small values of average body charges, "qba" and "qbs," in Coulomb scattering term were causing an overshoot (hump) in Slope Ratio Test[1] of a FinFET. To fix the problem these values were replaced by a constant, 1.0E-2/Cox. This change results in a correct weak-inversion I_D-V_G slop/behavior and passing Slope Ratio Test.

- **Lateral Non-uniform Doping (NUD)**

Lateral NUD Model was introduced to create *IV-CV* Vth shift. New parameters *K0* and *K0SI* were introduced. The parameters are binnable and temperature dependent.

In BSIM-CMG106.1.0

```
+ if(K0_i != 0) begin
+         T1     =  K0_t / (K0SI_t * qia + 2.0 * nVtm);
+         Mnud   =  lexp(-T1);
+ end else
+         Mnud   =  1.0;
```

<u>Drain current updated to include the new effects:</u>

```
- ids  =  NFINtotal * beta * ids0 * Moc / (Dmob * Dvsat * Dr);
+ ids  =  NFINtotal * beta * ids0 * Moc * Mob * Mnud / (Dmob * Dvsat * Dr);
```

[1] Xin Li *et al.*, "Benchmark Tests for MOSFET Compact Models With Application to the PSP Model," *Electron Devices, IEEE Transactions on* , vol.56, no.2, pp.243-251, Feb. 2009.

- **Body Effect for BULKMOD=1**

New parameters were introduced: *PHIBE*, *K1*, *K1SAT*, and *K1SI*. The parameters are binnable, and except for *PHIB*E, they are also temperature dependent.

In BSIM-CMG106.1.0

```
+ if(BULKMOD != 0) begin
+           vesx    =  ves - 0.5 * (vds - vdsx);
+           vesmax  =  0.95 * PHIBE_i;
+           T2      =  vesmax - vesx - 1.0E-3;
+           veseff  =  vesmax - 0.5 * (T2 + sqrt(T2 * T2 + 0.004 * vesmax));
+ end
---------------------------------
// *** Body- effect factor for BULKMOD = 1 ***
if(BULKMOD != 0) begin
            T0     =  hypsmooth((K1_t + K1SAT_t * vdsx),1.0E-6);
            T1     =  T0 / (K1SI_t * qia + 2.0 * nVtm);
            T3     =  sqrt(PHIBE_i - veseff) - sqrt(PHIBE_i);
            Mob    =  lexp(- T1 * T3);
end else
            Mob = 1.0;
```

<u>Drain current updated to include the new effects:</u>

```
- ids  =  NFINtotal * beta * ids0 * Moc / (Dmob * Dvsat * Dr);
+ ids  =  NFINtotal * beta * ids0 * Moc * Mob * Mnud / (Dmob * Dvsat * Dr);
```

<u>Added body-effect to mobility model with parameters UC and UC1:</u>

In BSIM-CMG106.0.0

```
- T3    =  UA_t * pow(abs(Eeffm), EU_i) + UD_t / T2;
```

In BSIM-CMG106.1.0

```
+ if(BULKMOD != 0) begin
+           T3    =  (UA_t + UC_t * veseff) * pow(abs(Eeffm), EU_i) + UD_t / T2;
+ end else begin
+           T3    =  UA_t * pow(abs(Eeffm), EU_i) + UD_t / T2;
+ end
```

- **Cannel Length Modulation (CLM) Model**

CLM Model for *I-V* was modified. <u>Parameter *VASAT* removed</u>. The old equation was known to cause unphysical wiggles.

In BSIM-CM-G106.0.0

```
- Mclm   =  1.0 + T1 * lln(1.0 + (vds - Vdseff) / T1 / VASAT_i);
```

In BSIM-CMG106.1.0

```
+ Mclm   =  1.0 + T1 * lln(1.0 + (vds - Vdseff) / T1 / (Vdsat + EsatL));
```

- **Velocity Saturation Model**

Velocity Saturation Model was modified for a better Id,sat and Gm,sat fitting. The new parameter *PSAT* (binnable & length scalable) was introduced.

In BSIM-CMG106.0.0

```
- T0     =  dqi / Esat1L;
- Dvsat  =  (1.0 + sqrt(DELTAVSAT + T0 * T0)) / (1.0 + sqrt(DELTAVSAT));
```

In BSIM-CMG106.1.0...

```
+ PSAT_i =  PSAT_i + APSAT * lexp(-Leff / BPSAT);
+ T0        =  lexp(PSAT_i * lln(dqi / Esat1L));
```

```
+ Ta        =  (1.0 + lexp(1.0/PSAT_i * lln(DELTAVSAT_i)));
+ Dvsat     =  (1.0 + lexp(1.0/PSAT_i * lln(DELTAVSAT_i + T0))) / Ta;
```

- **Non-saturation Effect**

Non-saturation effect was introduced with new parameters *A1* and *A2* to improve Id,sat and Gm,sat fitting.

In BSIM-CMG106.1.0

```
// ** Non-saturation effect **
+ T0      =  A1_t + A2_t / (qia + 2.0 * nVtm);
+ T1      =  T0 * dqi * dqi;
+ T2      =  T1 + 1.0 - 0.001;
+ T3      =  -1.0 + 0.5 * (T2 + sqrt(T2*T2 + 0.004));   //max(T1,-1.0)
+ Nsat    =  0.5 * (1.0 + sqrt(1.0 + T3));
+ Dvsat   =  Dvsat * Nsat;
```

- **Short Channel *C-V***

a) Velocity Saturation Model:

New parameter *VSATCV* (binnable) was added. To enhance the fitting accuracy, new parameters *VSATCV*, *DELTAVSATCV*, and *PSATCV* where introduced. The default values for these parameters are *VSAT*, *DELTAVSAT*, and *PSAT*, respectively. The parameters are binnable.

In BSIM-CMG106.0.0

```
- eta_mu    =  0.5 * ETAMOB_t;
- if( DEVTYPE != `ntype ) begin
-           eta_mu    =  1.0 / 3.0 * ETAMOB_t;
- end
```

In BSIM-CMG106.1.0

```
+ eta_mu    =  0.5 * ETAMOB_t;
+ eta_mu_cv =  0.5;
+ if( DEVTYPE != `ntype ) begin
+           eta_mu    =  1.0 / 3.0 * ETAMOB_t;
+           eta_mu_cv =  1.0 / 3.0;
+ end

// Mobility
+ Eeffm_cv =  EeffFactor * (qba + eta_mu_cv * qia);    // in the unit of MV/cm
+ T3       =  UA_t * pow(abs(Eeffm_cv), EU_i) + UD_t / T2;
+ Dmob_cv  =  1.0 + T3;
+ Dmob_cv  =  Dmob_cv / U0MULT;

// Velocity Saturation
// *** Velocity Saturation factor for C-V ***
EsatCV   =  2.0 * VSATCV_t * Dmob_cv / u0;
EsatCVL  =  EsatCV * LeffCV;
T0       =  lexp(PSATCV_i * lln(dqi / EsatCVL));
Ta       =  (1.0 + lexp(1.0/PSATCV_i * lln(DELTAVSATCV_i)));
DvsatCV  =  (1.0 + lexp(1.0/PSATCV_i * lln(DELTAVSATCV_i + T0))) / Ta;
```

b) CLM Model for *C-V* was modified:

I. New parameter *PCLMCV* (binnable) was introduced:

In BSIM-CMG106.1.0

```
+ if(PCLMCV_i != 0)
+      MclmCV =  1.0 + PCLMCV_i * lln(1.0 + (vds - Vdseff) / PCLMCV_i / (Vdsat + EsatCVL));
+ else
```

+ MclmCV = 1.0;

II. Charge equations in bsimcmg_quasi_static_cv.include were updated:

In BSIM-CMG106.0.0
- qd = NFINtotal * qd * T1;
- qg = NFINtotal * (qg * T1 - qb);

In BSIM-CMG106.1.0
+ inv_MclmCV = 1.0 / MclmCV;
+ qg = inv_MclmCV * qg + (MclmCV - 1.0) * qid;
+ qd = inv_MclmCV * inv_MclmCV * qd + 0.5 * (MclmCV - inv_MclmCV) * qid;
+ qd = NFINtotal * qd * T1;
+ qg = NFINtotal * (qg * T1 - qb);

c)

BSIM-CMG106.0.0 bsimgcmg_quasi_static_cv.include
- //idscv = idscv * Mclm / Dvsat / Mcv ; //Uncomment for Short Channel CV

BSIM-CMG106.1.0 bsimgcmg_quasi_static_cv.include
+ idscv = idscv / DvsatCV;

- **Junction Capacitance**

The equations were put into a macro and moved to commmon_def's. Junction capacitance equations have also been changed. The transition from standard junction capacitance equation to linear cap does not happen at *V*es =0 / *V*ed=0 (like BSIM4), but at *V*es = 0.9 * Vbi (built in voltage of diode). This moves discontinuity in second derivative of capacitance far into forward bias where the device does not usually operate.

- **Drain to Source Fringe Capacitance**

Drain to source fringe capacitance is now available for all CGEOMODs.

In BSIM-CMG106.0.0
- qds_fr = CDSP * V(d, s); // only for CGEOMOD=1

In BSIM-CMG106.1.0
+ qds_fr = CDSP * V(d, s); // for all CGEOMODs

- **Removing QM Effects in Fringe Capacitance from CGEOMOD=0**

In BSIM-CMG106.0.0
- if(CGEOMOD == 0) begin
- T1 = NFINtotal * WeffCV;

In BSIM-CMG106.1.0
+ if(CGEOMOD == 0) begin
+ T1 = NFINtotal * WeffCV0;

- **Gate to Substrate Overlap Capacitance**

Gate to Substrate Overlap capacitance scalability was improved. New parameter *CGBN* introduced**.**

In BSIM-CMG106.0.0
- cgbox = CGBO * LeffCV;

In BSIM-CMG106.1.0
+ cgbox = (CGBO * NF * NGCON + CGBN * NFINtotal) * LeffCV;

- **Accumulation Region Capacitance**

  The new parameter *DLCACC* was introduced for accumulation region capacitance (CAPMOD=1 and BULKMOD=1)

  In BSIM-CMG106.1.0
  + LeffCV_acc = LeffCV - DLCACC;

  Note) LeffCV_acc will instead be used for calculation of Qb,acc and Qg,acc.

- **RDSMOD=0**

  RDSMOD=0 is now similar to that of BSIM4.

  For *Vdsat* calculation:

  In BSIM-CMG106.0.0
  - Rdss = (RSourceGeo + RDrainGeo + (RDSWMIN_i + RDSW_i / (1.0 + PRWG_i * qis)) * WeffWRFactor) * (NFINtotal) * rdstemp;

  In BSIM-CMG106.1.0
  + T4 = 1.0 + PRWGS_i * qis;
  + Rdss = (RDSWMIN_i + RDSW_i * T0) * WeffWRFactor * NFINtotal * rdstemp;

  For Drain current calculation:

  In BSIM-CMG106.0.0
  - T4 = 1.0 + PRWG_i * qia;
  - Rdsi = rdstemp * (RSourceGeo + RDrainGeo + (RDSWMIN_i + RDSW_i * T0) * WeffWRFactor);

  In BSIM-CMG106.1.0
  + Rsource = RSourceGeo;
  + Rdrain = RDrainGeo;
  + T4 = 1.0 + PRWGS_i * qia;
  + Rdsi = rdstemp * (RDSWMIN_i + RDSW_i * T0) * WeffWRFactor;

  Note) Rsource and Rdrain are placed between nodes (si,s) and (di,d).

- **RDSMOD=1**

  RDSMOD=1 was enhanced to capture quasi-saturation/current crowding for high voltage devices. New parameters *RSDR*, *RDDR*, *PRSDR*, and *PRDDR* were introduced and new equations were added. *RSDR*, *RDDR* are temperature dependent through parameters *TRSDR* and *TRDDR*.

  In BSIM-CMG106.0.0
  // Rsource equation
  - T3 = sqrt(T2 * T2 + 1.0e-4);
  - Rsource = rdstemp * (RSourceGeo + (RSWMIN_i + RSW_i * T0) * WeffWRFactor);
  // Rdrain equation
  - T3 = sqrt(T2 * T2 + 1.0e-4);
  - Rdrain = rdstemp * (RDrainGeo + (RDWMIN_i + RDW_i * T0) * WeffWRFactor);

  In BSIM-CMG106.1.0
  + RSDR_t = RSDR * hypsmooth(1.0 + TRSDR * delTemp - 1.0E-6, 1.0E-3);
  + RDDR_t = RDDR * hypsmooth(1.0 + TRDDR * delTemp - 1.0E-6, 1.0E-3);
  // Rsource equation
  + T3 = sqrt(T2 * T2 + 1.0e-1); // Done to prevent glitch in Gds2 at vgs_eff =0
  + T5 = RSW_i* (1.0 + RSDR_a * lexp(0.5 * PRSDR * lln(V(si,s) * V(si,s) + 1.0E-6)));
  + Rsource = rdstemp * (RSourceGeo + (RSWMIN_i + T5 * T0) * WeffWRFactor);
  // Rdrain equation
  + T3 = sqrt(T2 * T2 + 1.0e-1); // Done to prevent glitch in Gds2 at vgd_eff =0

```
+ T5       =  RDW_i *(1.0 +  RDDR_a * lexp(0.5 * PRDDR * lln(V(di,d) * V(di,d) + 1.0E-6)));
+ Rdrain   =  rdstemp * (RDrainGeo + (RDWMIN_i + T5 * T0) * WeffWRFactor);
```

- **Asymmetric Model**

a) Following parameters were added to Asymmetric Model: *PDIBL1R*, *PTWGR*, *VSAT1R*, *RSDRR*, and *RDDRR*.

b) In Addition, parameter *PRWG* was split into *PRWGS* and *PRWGD* for source and drain sides in RDSMOD=1.

In BSIM-CMG106.0.0

```
// Source Resistance
- T4     =  1.0 + PRWG_i * vgs_eff;
// Drain Resistance
- T4     =  1.0 + PRWG_i * vgd_eff;
```

In BSIM-CMG106.1.0

```
// Source Resistance
+ T4     =  1.0 + PRWGS_i * vgs_eff;
// Drain Resistance
+ T4     =  1.0 + PRWGD_i * vgd_eff;
```

Note) For RDSMOD=0, PRWGS takes over and replaces PRWG. PRWGD is not used.

- **Gate Current Model**

a) The oxide thickness used in gate tunneling current

The gate tunneling current model in BSIM-CMG106.0.0 uses the physical oxide thickness *TOXP*. *TOXP* is also used in *C-V* model. In order to decouple the oxide thickness used in *I-V* model (through the gate current) and *C-V* model and facilitate the model tuning, a new parameter, *TOXG*, is introduced and used in the gate tunneling current model in BSIM-CMG106.1.0. The default value of *TOXG* is equal to *TOXP*.

b) The equations around BSIM4 parameters NTOX (binnable) and TOXREF were added.

In BSIM-CMG106.0.0

```
- Toxratio =  1.0 / (TOXP * TOXP);
- Toxratioedge =  Toxratio / (POXEDGE_i * POXEDGE_i);
```

In BSIM-CMG106.1.0

```
+ T0 = TOXG * TOXG;
+ T1 = TOXG * POXEDGE_i;
+ T2 = T1 * T1;
+ ToxRatio = lexp(NTOX_i * lln(TOXREF / TOXG))/ T0;
+ ToxRatioEdge = lexp(NTOX_i * lln(TOXREF / T1)) /T2;
```

- **Impact Ionization model**

Some cosmetic changes were made to Impact Ionization model (especially IIMOD=1), implementing smooth functions.

In BSIM-CMG106.0.0

```
- T0 =  BETAII2_i + BETAII1_i * Vdiff + BETAII0_i * Vdiff * Vdiff;
- if (T0 < 1e-5) begin
-    T0 =  1e-5;
- end
- if ((T0 < Vdiff / `EXPL_THRESHOLD) && (Vdiff > 0.0)) begin
-          Ratio  =  ALPHAII * `MAX_EXPL;
- end else if ((T0 < -Vdiff / `EXPL_THRESHOLD) && (Vdiff < 0.0)) begin
-          Ratio  =  ALPHAII * `MIN_EXPL;
- end else begin
```

```
-           Ratio  =  ALPHAII* lexp(Vdiff / T0);
- end
- /* Avoid too high ratio */
- if (Ratio > 10.0) begin
-           Ratio  =  10.0;
- end
```

In BSIM-CMG106.1.0

```
+ T0       =  BETAII2_i + BETAII1_i * Vdiff + BETAII0_i * Vdiff * Vdiff;
+ T1       =  sqrt(T0*T0 + 1.0E-10);
+ T2       =  10.0 - ALPHAII* lexp(Vdiff / T1) - 0.01;
+ /* Avoid too high ratio */
+ Ratio  =  10.0 - 0.5 * (T2 + sqrt(T2*T2 + 4.0*10.0*0.01)); //min(10,ALPHAII* lexp(Vdiff / T1));
```

- **Gate-induced source and drain leakage (GIDL/GISL)**
  GIDL/GISL current models were modified. New parameter *PGIDL*/*PGISL* introduced. The model now retains zero current at zero bias and it is smooth w.r.t. bias. For BULKMOD=1, the singularity in the denominator of body-bias factor, i.e., $Vde^3/(CGIDL+Vde^3)$, was removed and for $Vde>0$ the GIDL is zero (See Appendix A). Similar fixes were applied to GISL model.

<table>
<tr><th>BSIM-CMG106.0.0</th><th>BSIM-CMG106.1.0</th></tr>
<tr><td><pre>
// GIDL
T1 = (- vgd_noswap - EGIDL_i + vfbsd) / T0;
if ((T1 <= 0.0) || (AGIDL_i <= 0.0) || (BGIDL_t <= 0.0))
begin
   T5 = 0.0;
end else begin
    T2 = BGIDL_t / T1;
    if(BULKMOD != 0) begin
        T3 = - ved_jct*ved_jct*ved_jct;
        T4 = T3 / (CGIDL_i + T3);
    end else begin
        T3 = vds_noswap * vds_noswap;
        T4 = 0.5 * (vds_noswap + sqrt(T3 + 4.0E-4)) - 0.01;
    end
    if (T2 < 100.0)
        T5 = AGIDL_i * Weff0 * T1 * lexp(-T2) * T4;
    else T5 = AGIDL_i * Weff0 * 3.720075976e-44 * T4;
end
if(sigvds > 0.0) igidl = T5;
else igisl = T5;
</pre></td><td><pre>
// ** GIDL **
if ((AGIDL_i <= 0.0) || (BGIDL_t <= 0.0)) begin
    T6 = 0.0;
end else begin
    T1 = (- vgd_noswap - EGIDL_i + vfbsd) / T0;
    T1 = hypsmooth(T1, 1.0E-2);
    T2 = BGIDL_t / (T1 + 1.0E-3);
    T3 = lexp(PGIDL_i * lln(T1));
    if(BULKMOD != 0) begin
        T4   = - ved_jct*ved_jct*ved_jct;
        T4a = CGIDL_i + abs(T4) + 1.0E-9;
        T5   = hypsmooth(T4/T4a, 1.0E-6)-1.0E-6;
        T6 = AGIDL_i * Weff0 * T3 * lexp(-T2) * T5;
    end else
    T6 = AGIDL_i * Weff0 * T3 * lexp(-T2) * vds_noswap;
end
  if(sigvds > 0.0) igidl = T6;
  else igisl = T6;
</pre></td></tr>
</table>

- **Negative TNOM**

  Allowing default modelcards at negative TNOM:

In BSIM-CMG106.0.0

```
- parameter real TNOM      =  27.0 from [0:inf);
- if(TNOM <= 0) begin
-           $strobe("Warning: TNOM = %e is not positive. Set to 300.15 K.", TNOM);
-           Tnom = `REFTEMP;
- end else begin
-           Tnom = TNOM + `CONSTCtoK;
- end
```

In BSIM-CMG106.1.0

```
+ parameter real TNOM      =  27.0 from [ -`P_CELSIUS0:inf);
+ if(TNOM < -`P_CELSIUS0) begin
+           $strobe("Warning: TNOM < -`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
```

- **Temperature Dependence for *ETAMOB***
  The model parameter *EMOBT* now has linear temperature dependency:
  + ETAMOB_t = ETAMOB_i * hypsmooth(1.0 + EMOBT_i*delTemp - 1.0E-6, 1.0E-3);

- **Temperature Dependence for Subthreshold Swing**

  Temperature Dependence for Subthreshold Swing was enhanced with the new parameter *TSS* (binnable).

  In BSIM-CMG106.0.0
  - nVtm = Vtm * (1.0 + (CIT_i / TRatio + cdsc) / T1);

  In BSIM-CMG106.1.0
  + ThetaSS = hypsmooth(1.0 + TSS_i*delTemp - 1.0E-6, 1.0E-3);
  + nVtm = Vtm * ThetaSS * (1.0 + (CIT_i / TRatio + cdsc) / T1);

- **Temperature dependence of gate tunneling current**

  The model parameters *AIGBINV*, *AIGBACC*, *AIGC*, *AIGS*, and *AIGD* now have linear temperature dependency, e.g. *AIGBINV_t = AIGBINV_i + AIGBINV1_i* * (*T* - *TNOM*).

- **Temperature dependence of body effect model, lateral NUD model, and non-saturation effect model**

  The model parameters *K1*, *K1SAT*, *K0*, *K0SI*, *A1*, and *A2* now have linear temperature dependency, e.g. *K1_t = K1_i + K11_i* * (*T* - *TNOM*).

- **Decouple the model parameter relationship**

  In BSIM-CMG106.0.0 the temperature effect on U_0 was also affecting UTL:

  $$\mu_0(T) = U_0[L].\left(\frac{T}{TNOM}\right)^{UTE_i}.[1 + UTL_i.(T - TNOM)]. \quad \text{(old)}$$

  The original $\mu_0(T)$ equation was changed to

  $$\mu_0(T) = U_0[L].\left(\frac{T}{TNOM}\right)^{UTE_i} + UTL_i.(T - TNOM). \quad (3.69)$$

  Similarly, the original UA equation

  $$UA(T) = UA[L].[1 + UA1_i.(T - TNOM)] \quad \text{(old)}$$

  was changed to

  $$UA(T) = UA[L] + UA1_i.(T - TNOM). \quad (3.71)$$

- **Temperature and length dependence of impact ionization current**

  In impact ionization model, for IIMOD=2, $ALPHAII$ now has the following format:

  $$ALPHAII(T) = \frac{ALPHAII0(T) + ALPHAII1(T).L_{eff}}{L_{eff}},$$

  where $ALPHAII0$ and $ALPHAII1$ have linear temperature dependency.

- **Current (Is) clamping in accumulation**
  Imin is now a parameter with the default value of 1.0E-15.

- **Thermal noise, TNOIMOD=0**

  In BSIM-CMG106.0.0
  - T0 = 5.0 * nVtm * csi / cox;

```
- T1     =  qis * qis + qis * qid + qid * qid;
- if (COREMOD != 0) etatnoi  =  2.0;
- else    etatnoi  =  2.0 - (T0 + 2.0 * qbs) / (T0 + qia);
- idscv  =  qia + etatnoi * nVtm;
- qinv   =  cox * WeffCV * LeffCV * (T1 / 3.0 + etatnoi * nVtm * qia) / idscv;
```

Note) Instead qinv (channel charge) is calculated using the charge calculated for *C-V*.

In BSIM-CMG106.1.0 bsimcmg_quasi_static_cv.include

```
// *** Multiply Normalized capacitances by gate area ***
+ T1   =  WeffCV * LeffCV * coxeff;
+ qinv =  NFINtotal * qg * T1;       //Also used for TNOIMOD=0
+ qd   =  NFINtotal * qd * T1;
+ qg   =  qinv - NFINtotal * qb;
```

- **Built-in cosh() overflow**

When the argument *x* is too huge, the built-in *cosh(x)* could cause arithmetic point error due to overflow. To protect the code, for x > 40, 0.5/(*cosh(x)*-1) was replaced by *exp(-x)* as follows:

```
- Theta_SCE  =  0.5 / (cosh(DVT1_i * Leff / scl + 1e-6) - 1.0);

+ tmp = DVT1_i * Leff / scl + 1.0e-6;
+ if (tmp < 40.0)
+     Theta_SCE  =  0.5 / (cosh(tmp) - 1.0);
+ else
+     Theta_SCE  = exp(-tmp);
```

```
- Theta_DIBL  =  0.5 / (cosh(DSUB_i * Leff / scl + 1e-6) - 1.0);

+ tmp = DSUB_i * Leff / scl + 1e-6;
+ if (tmp < 40.0)
+     Theta_DIBL  =  0.5 / (cosh(tmp) - 1.0);
+ else
+     Theta_DIBL  =  exp(-tmp);
```

```
- DIBLfactor   =  0.5 * PDIBL1_a / (cosh(DROUT_i * Leff/scl + 1e-6) - 1.0) +
PDIBL2_i;

+ tmp = DROUT_i * Leff/scl + 1.0e-6;
+ if (tmp < 40.0)
+     DIBLfactor   =  0.5 * PDIBL1_a / (cosh(tmp) - 1.0) + PDIBL2_i;
 + else
 +     DIBLfactor   =  PDIBL1_a * exp(-tmp); + PDIBL2_i;
```

- **Threshold voltage definition**

A simple analytical V_{TH} definition was derived and implemented as operating point info in this version.

1. For a long channel device, the threshold voltage V_{TH} is defined as the value of V_G at which the drift and diffusion components of the source to drain current <u>at the source side</u> are equal.
2. Based on the definition in Step 1, it can be shown that at $V_G = V_{TH}$: $Q_{is} = C_{ox} V_{tm}$.
3. From Gauss's Law $V_G = V_{FB} + \psi_s + q_{is} + q_{bs}$.

4. $\psi_s \approx V_{tm}\ln\left\{\frac{Q_{is}\ (Q_{is}+2Q_{bulk}+5C_{Si}V_{tm})}{2qn_i\varepsilon_{sub}V_{tm}}\right\} + \varphi_B + \Delta V_{t,QM}$.[2]
5. For a long channel device, $V_{TH} = V_{FB} + V_{tm}\ln\left\{\frac{C_{ox}V_{tm}(C_{ox}V_{tm}+2Q_{bulk}+5C_{Si}V_{tm})}{2qn_i\varepsilon_{sub}V_{tm}}\right\} + \varphi_B + \Delta V_{t,QM} + V_{tm} + q_{bs}$.
6. Corrections due to threshold voltage roll-off, DIBL, reverse short channel effect, and temperature are added.
7. *Verilog-A Code:*

```
q0    =  (5.0 * Vtm * (epssub/TFIN) + 2.0 * Qbul) / cox;
T1    =  Vtm * (Vtm + q0);
T2    =  cox * cox * T1;
T3    =  2.0 * `q * ni * epssub * Vtm;
VTH   =  VFB + Vtm * lln(T2/T3) + phib + dvch_qm + Vtm + qbs + dvth_all - DELVTRAND;
```

• Appendix A

Gate-induced source and drain leakage

As illustrated in Fig. 1, for *GIDL/GISL* current to follow, the band bending at the gate-to-drain/source overlap region should be greater than or equal to the energy band gap *Eg* of silicon; the condition is required to level the valance band and conduction band so that the quantum mechanical tunneling process could take place.

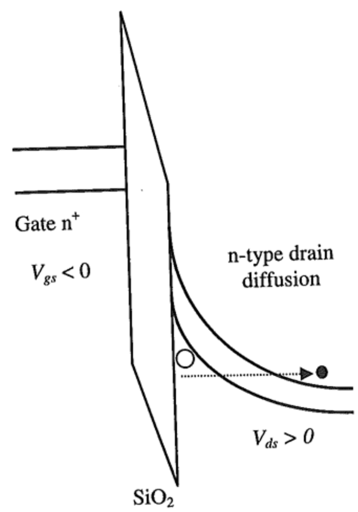

Fig.1. Schematic illustration of the energy band diagram; the band-to-band tunneling process in the gate/drain overlap region triggers for a band bending ≥ Eg. Electron-hole pairs are generated by the tunneling of valence band electrons into the conduction band and collected by the drain and body, respectively.

To get a band bending greater than or equal to *Eg*, the gate-to-drain/source overlap region should enter the *deep-depletion* operating mode. Considering an *n*-MOSFET for the sake of discussion, to enter the deep-depletion, the holes generated at the gate-to-drain/source overlap region should be collected by a negatively (relative to drain/source) biased body (*V*de or *V*se > 0). For *V*de or *V*se < 0, the holes remain at the interface and an inversion layer forms, where the band bending is pinched roughly at $2\Phi_B$, a value smaller than *Eg*. As a result, for *V*de or *V*se < 0, *GIDL/GISL* current is zero. For more details, please refer to [3].

[2] M. V. Dunga, "Nanoscale CMOS Modeling," Ph.D. dissertation, BSIM Group, Univ. California, Berkeley, 2008, ch. 3, p. 66. Available online at: http://www.eecs.berkeley.edu/Pubs/TechRpts/2008/EECS-2008-20.pdf
[3] T.Y. Chan, J. Chen, P. K. Ko, and C. Hu, "The impact of gate-induced drain leakage current on MOSFET scaling," *Tech. Digest of IEDM*, pp. 718-721, 1987.