Summary of Changes from BSIM-BULK107.1.0 to BSIM-BULK107.2.0 Beta0_1:

BSIM Group, IIT Kanpur, UC Berkeley

Yawar Hayat Zarkob (yawarz21@iitk.ac.in), Ayushi Sharma (sayushi@iitk.ac.in),
Chetan Kumar Dabhi (dabhi@berkeley.edu), Girish Pahwa (pahwa@berkeley.edu)

A. Summary of Enhancements:

1. **2022enh2 (Infineon):** Accuracy enhancement in Impact Ionization model of Intrinsic MOSFET.
2. **2022enh3 (Infineon):** Accuracy enhancement in Impact Ionization model of Drift Region.
3. **2022enh4 (Infineon):** Modeling of conductivity modulation (Expansion) effect.
4. **2022enh5 (Infineon):** To improve reverse V_{ds} impact ionization current.
5. **2023enh1 (GF):** Introduction of MULT parameters (MULT_I, MULT_Q, MULT_FN).

B. Summary of bug-fixes:

6. **2022bug9 (IITK/UCB)**: Correction in substrate current flow.
7. **2022bug10 (ADI):** Better implementation of ln_one_plus_exp.
8. **2022bug11 (ADI):** Code cleaning.
9. **2022bug12 (TSMC):** Negative Cgd in BSIM_BULK.
10. **2023bug1 (IITK):** Typo in the bulk charge expression in manual.
11. **2023bug2 (ADI):** Noise QA test results have values less than simulator tolerance.
12. **2023bug3(IITK/UCB):** Correction in exponent factor of Electric field expression in drift region.

➢ DESCRIPTION OF ENHANCEMENTS:

1. 2022enh2 (Infineon): Accuracy enhancement in Impact Ionization model of Intrinsic MOSFET.

- The Impact Ionization current in Intrinsic MOSFET for different Vd and Vb is not captured by existing model for high voltages.
- Impact Ionization Model in intrinsic MOSFET in BSIM-BULK107.1.0 is:

$$\boxed{I_{ii} = ALPHA0\_i * I_{ds} * diffvds * exp\left(-\frac{BETA0\_t}{diffvds}\right)}$$

- In BSIM-BULK107.1.0, the equation of Vdseff was given by:

$$\boldsymbol{Vdseff = V_{ds} * \left(1 + \left(\frac{V_{ds}}{(Vdssat)}\right)^{\frac{1}{DELTA}}\right)^{-DELTA}}$$

- In BSIM-BULK107.2.0 Beta0_1, the above equation is modified as follows:

$$\boldsymbol{Vdseffii = V_{ds} * \left(1 + \left(\frac{V_{ds}}{((1+BETA1*Vds)*Vdssat)}\right)^{\frac{1}{DELTA}}\right)^{-DELTA}}$$

- BETA0_t is replaced with BETA0_eff (V_d dependent).

$$BETA0\_eff = \frac{BETA0\_t}{2} * \left(1 + \boldsymbol{Vdseffii}^{\boldsymbol{BETA2}}\right)$$

- To account for body bias dependency, ALPHA0_i is replaced with ALPHA0_eff.

$$ALPHA0\_eff = ALPHA0\_i * \left(1 + \boldsymbol{ALPHA1} * Vbsx + \boldsymbol{ALPHA2} * Vbsx^2\right)$$

$$\boxed{I_{ii} = ALPHA0\_eff * I_{ds} * diffvdsii * exp\left(-\frac{BETA0\_eff}{diffvdsii^{\boldsymbol{BETA3}}}\right)}$$

- For backward compatibility, following default values of the new parameters are used: BETA1=0; BETA2=0; BETA3=1; ALPHA1=0 ALPHA2=0.

- Impact Ionization model in intrinsic MOSFET in BSIM-BULK107.2.0 Beta0_1:

```
If (HVMOD==0)
{     Existing model
}
If (HVMOD==1)
{
Updated model  (correction for the Vd dependence at internal node):
      $\boxed{I_{ii} = ALPHA0\_eff * I_{ds} * diffvdsii * exp\left(-\frac{BETA0\_eff}{diffvdsii^{BETA3}}\right)}$

  }
```

2. 2022enh3 (Infineon): Accuracy enhancement in Impact Ionization model of Drift Region.

- Impact Ionization current in drift region for different Vd and Vb is not captured by BSIM-BULK107.1.0 model which is given below:

$$VDDROP = V(d,bi) - V_{dseff} - DRII2$$

$$\boxed{\boldsymbol{I_{subDR} = ALPHADR * VDDROP * I_{ds} * exp(\frac{-BETADR}{E_m})}}$$

- In BSIM-BULK 107.2.0 Beta0_1, the VDDROP equation is modified as follows:

$$VDDROP = V(d,s) - \mathbf{DRII3} * V_{dseffii} - DRII2 - \mathbf{CMD1} * V_{bcm}^{\mathbf{DRII4}}$$

$$\boxed{V_{bcm} = \sqrt{V(bi,b)^2 + 1e - 6}}$$

- To account for body bias dependency, ALPHADR is replaced with ALPHADR_eff.

$$ALPHADR\_eff = ALPHADR * \ (1 + \boldsymbol{ALPHADR1} * Vbsx \ + \ \boldsymbol{ALPHADR2} * Vbsx^2\ )$$

$$\boxed{I_{subDR} \ = ALPHADR\_eff \ * E_m * I_{ds} * \ exp(\frac{-BETADR}{E_m})}$$

Note: In BSIM-BULK107.1.0 there was a typo, *V(d,bi)* was present in the VDDROP equation. Now, in BSIM-BULK107.2.0 Beta0_1 it is replaced with *V(d,s).*

3. 2022enh4 (Infineon): Modeling the conductivity modulation (Expansion) effect.

- Kirk effect leads to high electron injection in the drift region.
- Charge density = NDRIFTD + extra charge concentration
- In BSIM-BULK107.2.0 Beta0_1 $I_{drift,sat\ D}$, is given as below:

$$\boxed{I_{drift,sat\ D} \ = NDRIFTD\ W.NF.VDRIFT_{eff} * (1 + \mathbf{CMD1} * V_{bcm}^{\mathbf{CMD2}})}$$

$$\boxed{V_{bcm} = \sqrt{V(bi,b)^2 + 1e - 6}}$$

$$V_{drift,sat\ D} \ = I_{drift,sat\ D} * R_0$$

Similarly,

$$R_{drift,sat\ D} = R_0 \left[1 + \delta_V \left(\frac{|V_{di1,di}|}{V_{drift,sat\ D}}\right)^{\beta}\right]^{\frac{1}{\beta}}$$

$$\boxed{I_{drift,sat\ s} = NDRIFTS\ W.NF.VDRIFT_{eff} * (1 + \boldsymbol{CMS1} * V_{bcm}^{\boldsymbol{CMS2}})}$$

- Updated drift region Impact Ionization model was used for this:

$$VDDROP = V(d,s) - \boldsymbol{DRII3} * V_{dseffii} - DRII2 - \boldsymbol{CMD1} * V_{bcm}{}^{\boldsymbol{DRII4}}$$

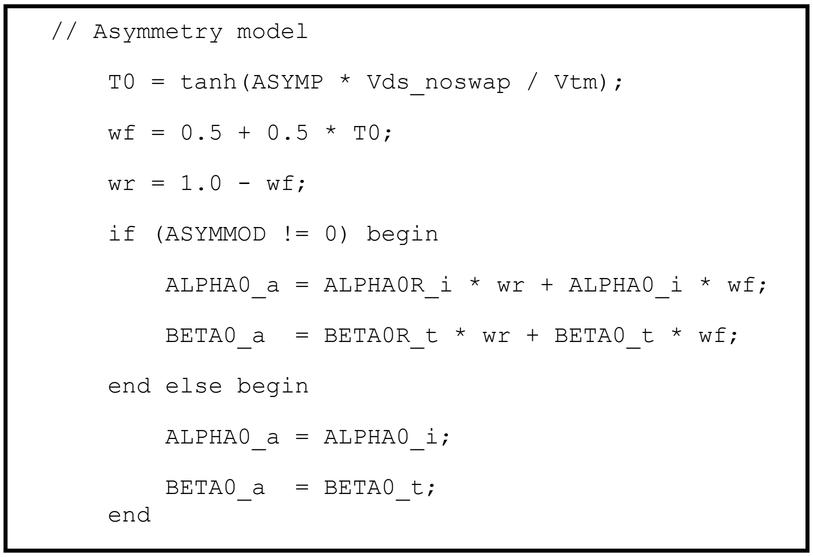
$$E_m = \left[\left(\frac{2q * N_{extra}}{\epsilon}\right) * VDDROP\right]^{0.5}$$

(Refer to 2023bug3 for modified E_m expression)

$$I_{subDR} = ALPHADR\_eff * E_m * I_{ds} * exp(\frac{-BETADR\_eff}{E_m})$$

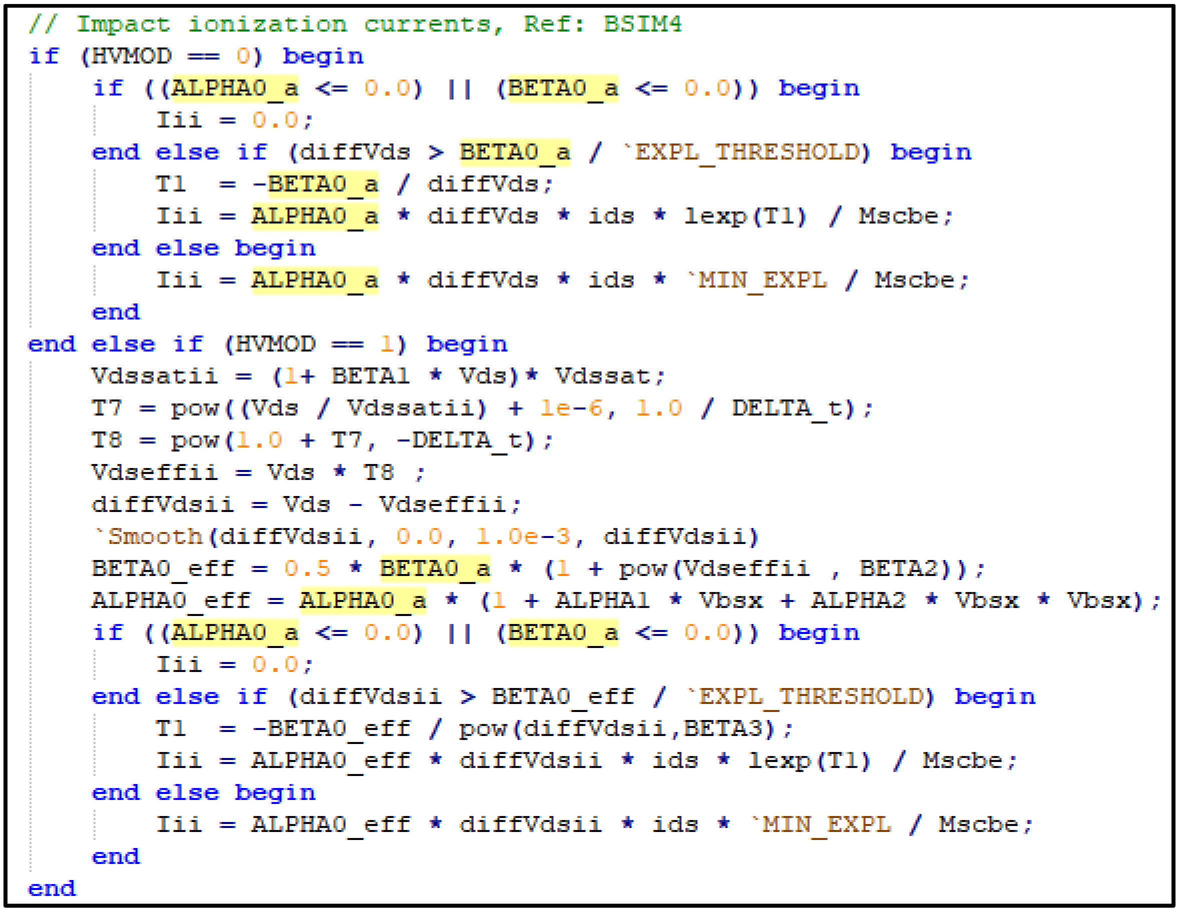
4. 2022enh5 (Infineon): To improve reverse V_{ds} impact ionization current.

- In reverse mode ($V_{DS} < 0$), impact ionization current is an order of magnitude higher due to the absence of LDD.
- In BSIM-BULK107.1.0, similar magnitude for impact ionization current in intrinsic MOSFET is observed for both directions of bias. Therefore, in BSIM-BULK 107.2.0 Beta0_1 ALPHA0R and BETA0R parameters are introduced separately under ASYMMOD as shown:



```
// Asymmetry model
    T0 = tanh(ASYMP * Vds_noswap / Vtm);
    wf = 0.5 + 0.5 * T0;
    wr = 1.0 - wf;
    if (ASYMMOD != 0) begin
        ALPHA0_a = ALPHA0R_i * wr + ALPHA0_i * wf;
        BETA0_a  = BETA0R_t * wr + BETA0_t * wf;
    end else begin
        ALPHA0_a = ALPHA0_i;
        BETA0_a  = BETA0_t;
    end
```

- The Verilog-A implementation of updated impact ionization model in intrinsic MOSFET after including this enhancement reads following in BSIM-BULK107.2.0 Beta0_1 now:



```
// Impact ionization currents, Ref: BSIM4
if (HVMOD == 0) begin
    if ((ALPHA0_a <= 0.0) || (BETA0_a <= 0.0)) begin
        Iii = 0.0;
    end else if (diffVds > BETA0_a / `EXPL_THRESHOLD) begin
        T1  = -BETA0_a / diffVds;
        Iii = ALPHA0_a * diffVds * ids * lexp(T1) / Mscbe;
    end else begin
        Iii = ALPHA0_a * diffVds * ids * `MIN_EXPL / Mscbe;
    end
end else if (HVMOD == 1) begin
    Vdssatii = (1+ BETA1 * Vds)* Vdssat;
    T7 = pow((Vds / Vdssatii) + 1e-6, 1.0 / DELTA_t);
    T8 = pow(1.0 + T7, -DELTA_t);
    Vdseffii = Vds * T8 ;
    diffVdsii = Vds - Vdseffii;
    `Smooth(diffVdsii, 0.0, 1.0e-3, diffVdsii)
    BETA0_eff = 0.5 * BETA0_a * (1 + pow(Vdseffii , BETA2));
    ALPHA0_eff = ALPHA0_a * (1 + ALPHA1 * Vbsx + ALPHA2 * Vbsx * Vbsx);
    if ((ALPHA0_a <= 0.0) || (BETA0_a <= 0.0)) begin
        Iii = 0.0;
    end else if (diffVdsii > BETA0_eff / `EXPL_THRESHOLD) begin
        T1  = -BETA0_eff / pow(diffVdsii,BETA3);
        Iii = ALPHA0_eff * diffVdsii * ids * lexp(T1) / Mscbe;
    end else begin
        Iii = ALPHA0_eff * diffVdsii * ids * `MIN_EXPL / Mscbe;
    end
end
```

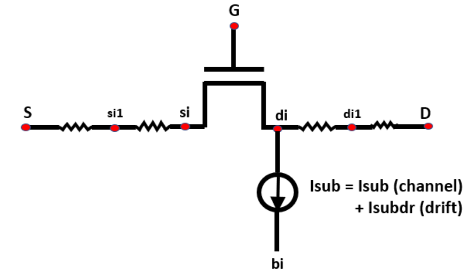
5. 2023enh1 (GF): Introduction of MULT parameters (MULT_I, MULT_Q, MULT_FN).

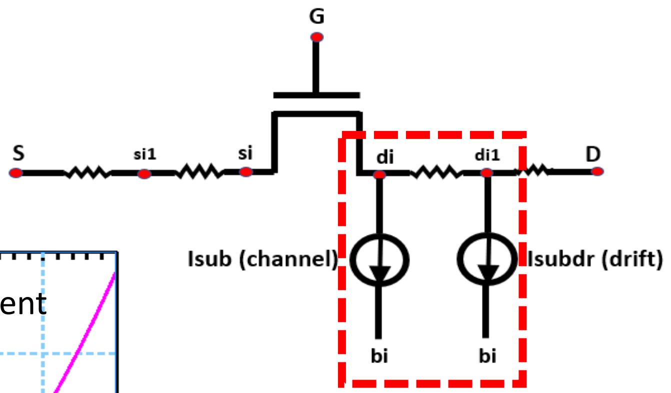
- MULT parameters are added in BSIM-BULK107.2.0 Beta0_1.
- MULT_I, MULT_Q and MULT_FN will account for variability in current, charges and flicker noise respectively.
- The default value of MULT factors is 1, So, it is a backward compatible change.

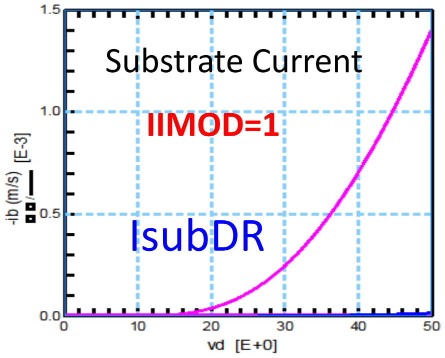
➢ DESCRIPTION OF BUGS-FIXES:

6. 2022bug9 (IITK/UCB): Correction in substrate current flow.

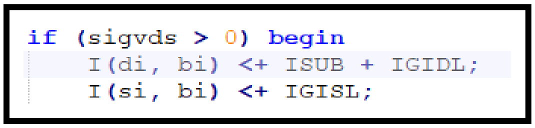
- In the BSIM-BULK107.1.0, both the components of substrate current (channel and drift region) flows through the internal node di. This causes the channel current to reduce and there is no change in the drain terminal current.
- In BSIM-BULK107.2.0 Beta0_1, the two components of substrate current have been separated. The channel substrate current flows from internal di and the drift region substrate current flows through the di1 node and the impact ionization current from the drift region is added to the drain terminal.





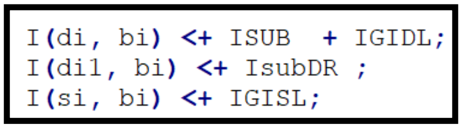


BSIM-BULK 107.1.0

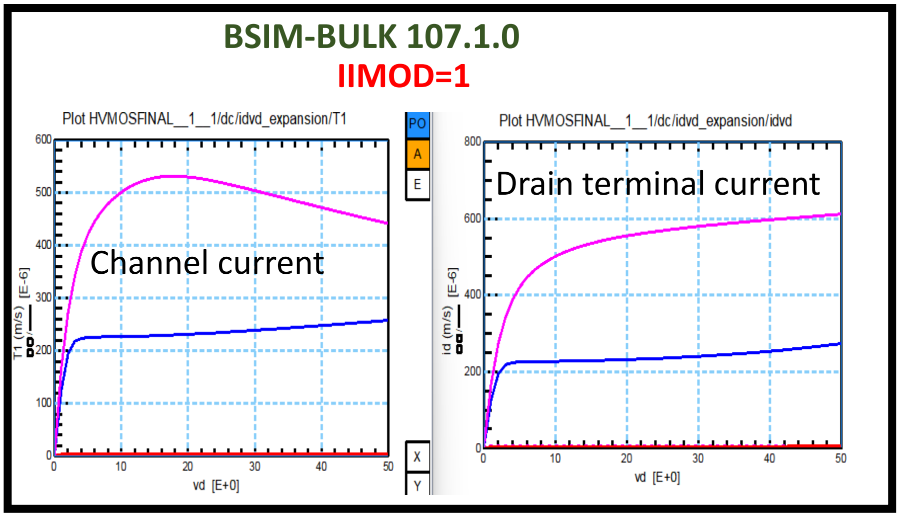


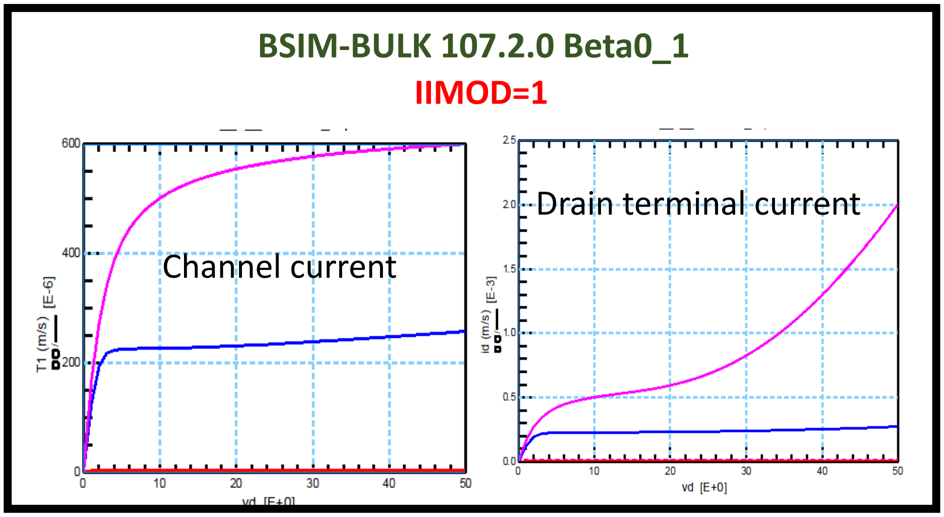
```
if (sigvds > 0) begin
    I(di, bi) <+ ISUB + IGIDL;
    I(si, bi) <+ IGISL;
```

BSIM-BULK 107.2.0 Beta0_1



```
I(di, bi) <+ ISUB  + IGIDL;
I(di1, bi) <+ IsubDR ;
I(si, bi) <+ IGISL;
```

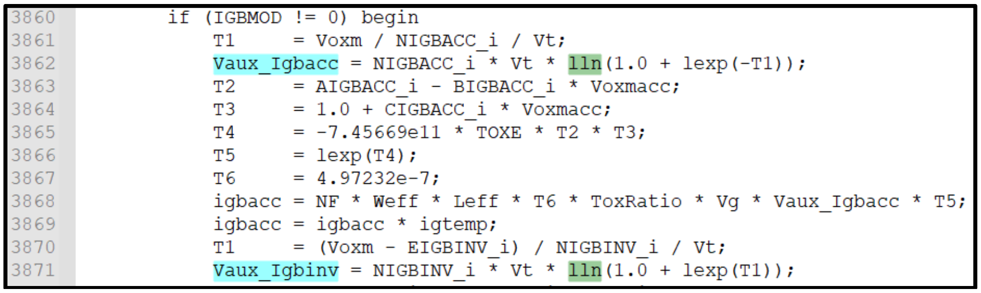



BSIM-BULK 107.2.0 Beta0_1

IIMOD=1

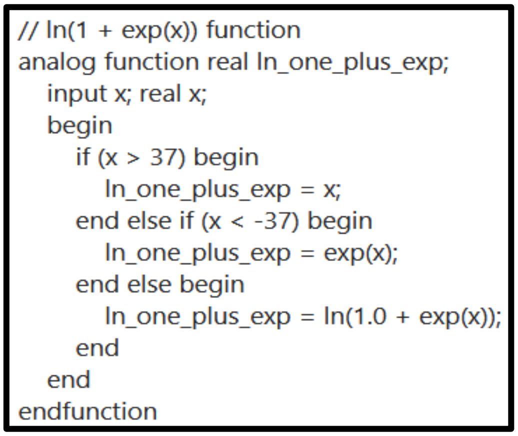
7. 2022bug10 (ADI): Better implementation of ln_one_plus_exp.

- There was no need to protect ln(x) with “lln” function in the following highlighted expressions because arguments will never be smaller than 1.



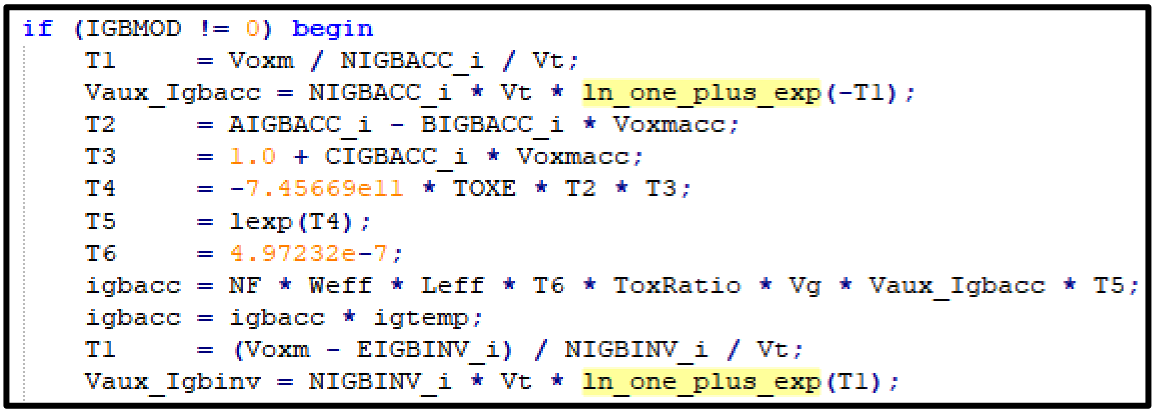
```
        if (IGBMOD != 0) begin
            T1      = Voxm / NIGBACC_i / Vt;
            Vaux_Igbacc = NIGBACC_i * Vt * lln(1.0 + lexp(-T1));
            T2      = AIGBACC_i - BIGBACC_i * Voxmacc;
            T3      = 1.0 + CIGBACC_i * Voxmacc;
            T4      = -7.45669e11 * TOXE * T2 * T3;
            T5      = lexp(T4);
            T6      = 4.97232e-7;
            igbacc = NF * Weff * Leff * T6 * ToxRatio * Vg * Vaux_Igbacc * T5;
            igbacc = igbacc * igtemp;
            T1      = (Voxm - EIGBINV_i) / NIGBINV_i / Vt;
            Vaux_Igbinv = NIGBINV_i * Vt * lln(1.0 + lexp(T1));
```

- For large values of T1, exp(T1) will dominate over 1.0 in double-precision arithmetic, such that when T1 > 37, ln(1.0 + exp(T1)) is numerically exactly x. So, in the second highlighted equation above calling of ln() and exp() can be avoided.
- We adopt following new function ln_one_plus_exp() with following functional implementation in BSIM-BULK107.2.0 Beta0_1:



```
// ln(1 + exp(x)) function
analog function real ln_one_plus_exp;
   input x; real x;
   begin
      if (x > 37) begin
         ln_one_plus_exp = x;
      end else if (x < -37) begin
         ln_one_plus_exp = exp(x);
      end else begin
         ln_one_plus_exp = ln(1.0 + exp(x));
      end
   end
endfunction
```

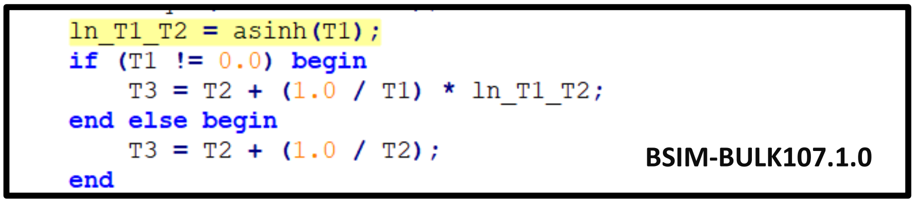
- The updated code reads as follows:



```
if (IGBMOD != 0) begin
    T1       = Voxm / NIGBACC_i / Vt;
    Vaux_Igbacc = NIGBACC_i * Vt * ln_one_plus_exp(-T1);
    T2       = AIGBACC_i - BIGBACC_i * Voxmacc;
    T3       = 1.0 + CIGBACC_i * Voxmacc;
    T4       = -7.45669e11 * TOXE * T2 * T3;
    T5       = lexp(T4);
    T6       = 4.97232e-7;
    igbacc = NF * Weff * Leff * T6 * ToxRatio * Vg * Vaux_Igbacc * T5;
    igbacc = igbacc * igtemp;
    T1       = (Voxm - EIGBINV_i) / NIGBINV_i / Vt;
    Vaux_Igbinv = NIGBINV_i * Vt * ln_one_plus_exp(T1);
```

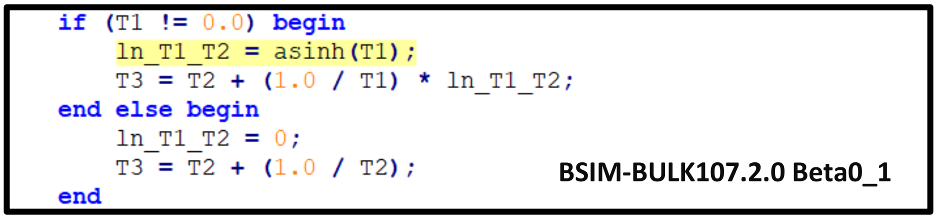
8. 2022bug11 (ADI): Code cleaning.

- The value of ln_T1_T2 is never used when T1 is zero therefore ln_T1_T2 is moved inside the if/else block:



```
ln_T1_T2 = asinh(T1);
if (T1 != 0.0) begin
    T3 = T2 + (1.0 / T1) * ln_T1_T2;
end else begin
    T3 = T2 + (1.0 / T2);
end
```

BSIM-BULK107.1.0



```
if (T1 != 0.0) begin
    ln_T1_T2 = asinh(T1);
    T3 = T2 + (1.0 / T1) * ln_T1_T2;
end else begin
    ln_T1_T2 = 0;
    T3 = T2 + (1.0 / T2);
end
```

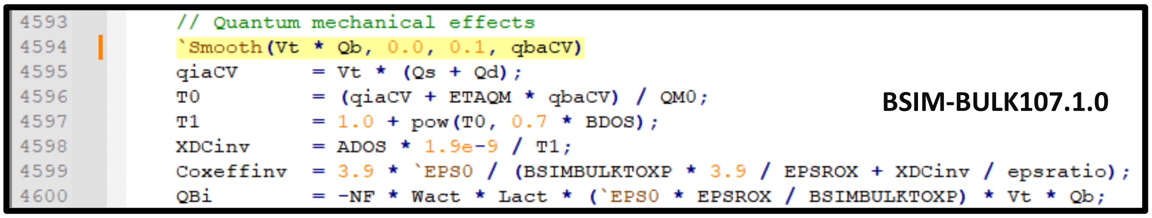
BSIM-BULK107.2.0 Beta0_1

9. 2022bug12 (TSMC): Negative Cgd in BSIM_BULK.

- Cgd was negative when CF=0 and ADOS=20 in CVMOD=1.

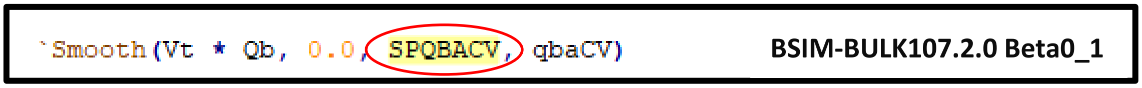
- It was observed for higher ADOS values smoothing of qbaCV made Cgd negative. So, a new parameter (**SPQBACV**) is introduced for flexibility which is a smoothing parameter for qbaCV.

- The default value of this new parameter is 0.1 which ensures the change is backward compatible.



```
// Quantum mechanical effects
`Smooth(Vt * Qb, 0.0, 0.1, qbaCV)
qiaCV       = Vt * (Qs + Qd);
T0          = (qiaCV + ETAQM * qbaCV) / QM0;
T1          = 1.0 + pow(T0, 0.7 * BDOS);
XDCinv      = ADOS * 1.9e-9 / T1;
Coxeffinv   = 3.9 * `EPS0 / (BSIMBULKTOXP * 3.9 / EPSROX + XDCinv / epsratio);
QBi         = -NF * Wact * Lact * (`EPS0 * EPSROX / BSIMBULKTOXP) * Vt * Qb;
```

BSIM-BULK107.1.0



```
`Smooth(Vt * Qb, 0.0, SPQBACV, qbaCV)
```

BSIM-BULK107.2.0 Beta0_1

10. 2023bug1 (IITK): Typo in the bulk charge expression in manual.

From the BSIM-BULK107.1.0 manual:

Bulk charge with poly depletion effect :

$$q_B = A + B + \frac{1}{3}\cdot\frac{\Delta q^2}{C^3}\cdot\left[\frac{4}{8}\cdot\left(C^2 + P.Q\right)\cdot\frac{1}{1+q_s+q_d} + \frac{2}{\gamma_g^2}\right] - n_q\cdot\left[q_s + q_d + \frac{1}{3}\cdot\frac{(q_s-q_d)^2}{1+q_s+q_d}\right] \tag{9.20}$$

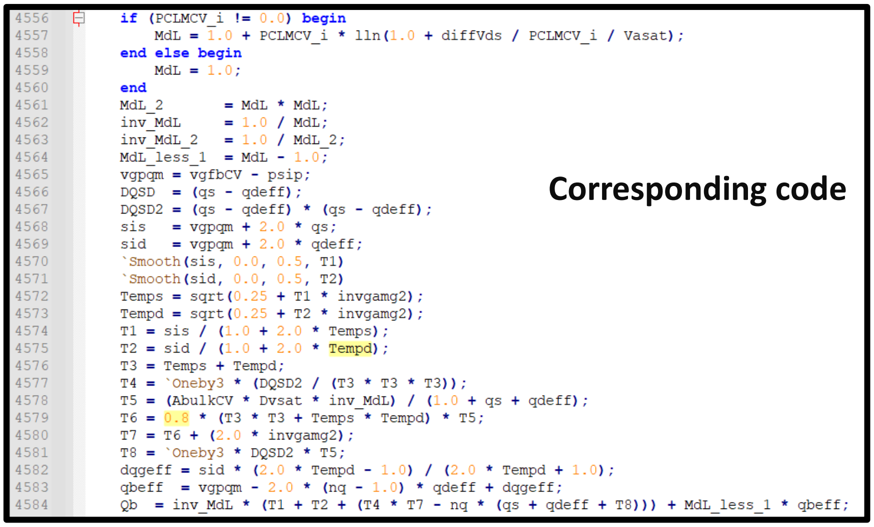
Typo: It should be $\frac{4}{5}$

$$A = \frac{v_g - v_{fb} - \psi_p + 2.q_s}{1 + 2.\sqrt{\frac{1}{4} + \frac{v_g - v_{fb} - \psi_p + 2.q_s}{\gamma_g^2}}} \tag{9.23}$$

$$B = \frac{v_g - v_{fb} - \psi_p + 2.q_d}{1 + 2.\sqrt{\frac{1}{4} + \frac{v_g - v_{fb} - \psi_p + 2.q_s}{\gamma_g^2}}} \tag{9.24}$$

Typo: It should be q_d

- This is also evident from the BSIM-BULK107.1.0 Verilog-A code:



```
    if (PCLMCV_i != 0.0) begin
        MdL = 1.0 + PCLMCV_i * lln(1.0 + diffVds / PCLMCV_i / Vasat);
    end else begin
        MdL = 1.0;
    end
    MdL_2       = MdL * MdL;
    inv_MdL     = 1.0 / MdL;
    inv_MdL_2   = 1.0 / MdL_2;
    MdL_less_1  = MdL - 1.0;
    vgpqm = vgfbCV - psip;
    DQSD  = (qs - qdeff);
    DQSD2 = (qs - qdeff) * (qs - qdeff);
    sis   = vgpqm + 2.0 * qs;
    sid   = vgpqm + 2.0 * qdeff;
    `Smooth(sis, 0.0, 0.5, T1)
    `Smooth(sid, 0.0, 0.5, T2)
    Temps = sqrt(0.25 + T1 * invgamg2);
    Tempd = sqrt(0.25 + T2 * invgamg2);
    T1 = sis / (1.0 + 2.0 * Temps);
    T2 = sid / (1.0 + 2.0 * Tempd);
    T3 = Temps + Tempd;
    T4 = `Oneby3 * (DQSD2 / (T3 * T3 * T3));
    T5 = (AbulkCV * Dvsat * inv_MdL) / (1.0 + qs + qdeff);
    T6 = 0.8 * (T3 * T3 + Temps * Tempd) * T5;
    T7 = T6 + (2.0 * invgamg2);
    T8 = `Oneby3 * DQSD2 * T5;
    dqgeff = sid * (2.0 * Tempd - 1.0) / (2.0 * Tempd + 1.0);
    qbeff  = vgpqm - 2.0 * (nq - 1.0) * qdeff + dqgeff;
    Qb  = inv_MdL * (T1 + T2 + (T4 * T7 - nq * (qs + qdeff + T8))) + MdL_less_1 * qbeff;
```

Corresponding code

- In the BSIM-BULK107.2.0 Beta0_1 these typos have been corrected in the manual.
- Additionally, the derivation is also included in the manual.

11. 2023bug2 (ADI): Noise QA test results have values less than simulator tolerance.

- Most of the N(g) noise tests had QA results < 10^{-30}.
- Since the simulator tolerance is 10^{-30} , the simulator assumes it as zero and passes the test.
- The following parameters are updated in the following respective tests in the qaSpec file:

Test 032_Noise1_WL:

```
instanceParameters      W=10.0e-3  L=0.1e-6


modelParameters         TOXE =1e-9
```

Test 034 _Noise3_WL:

```
instanceParameters      W=10.0e-4  L=0.1e-6


modelParameters         TOXE =1e-9
```

Existing: W=10.0e-6, L=1e-6, TOXE = 3e-9

12. 2023bug3 (IITK/UCB): Correction in exponent factor of Electric field expression in drift region (E_m).

- In BSIM-BULK107.1.0, the expression for E_m was given by:

$$\boxed{E_m = \left(\frac{2q * N_{extra}}{\epsilon}\right) * VDDROP}$$

- In BSIM-BULK107.2.0 Beta0_1, the expression for E_m is corrected as follows:

$$\boxed{E_m = \left[\left(\frac{2q * N_{extra}}{\epsilon}\right) * VDDROP\right]^{0.5}}$$