Summary of Changes from BSIM-BULK 107.1.0 to BSIM-BULK 107.2.0:

BSIM Group, IIT Kanpur, UC Berkeley

Yawar Hayat Zarkob (yawarz21@iitk.ac.in), Ayushi Sharma, Dinesh Rajasekharan, Chetan Kumar Dabhi, Ahtisham Pampori

BSIM-BULK 107.2.0_beta0_1

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:

1. **2022bug9 (IITK/UCB)**: Correction in substrate current flow.
2. **2022bug10 (ADI):** Better implementation of ln_one_plus_exp.
3. **2022bug11 (ADI):** Code cleaning.
4. **2022bug12 (TSMC):** Negative Cgd in BSIM_BULK.
5. **2023bug1 (IITK):** Typo in the bulk charge expression in manual.
6. **2023bug2 (ADI):** Noise QA test results have values less than simulator tolerance.
7. **2023bug3 (IITK/UCB):** Correction in exponent factor of Electric field expression in drift region.

BSIM-BULK 107.2.0_beta0_2

C. Summary of enhancements:

1. **2023enh2 (GF):** Flicker noise model enhancement.
2. **2023enh3 (GF):** Decoupling drain-side drift resistance for capacitance calculations.

D. Summary of bug-fixes:

1. **2023bug4 (ADI):** Issue in CGD reciprocity while exercising QM parameters.
2. **2023bug5 (ADI):** MULT implementation correction.
3. **2023bug6 (ADI):** Correction in units of binning parameters.
4. **2023bug7 (ADI):** Removing variables that were superfluously assigned.
5. **2023bug8 (ADI):** Addressing the bias-dependent $strobe warnings reported by VAMPyRE.

BSIM-BULK 107.2.0_beta1

E. Summary of bug-fixes:

1. **2023bug9 (ADI):** Removing the redundant 0V source in TNOIMOD = 0.

BSIM-BULK 107.2.0_beta2

F. Summary of enhancements:

1. **2023enh4 (ADI):** Fitting flexibility of Cgg in moderate inversion.
2. **2023enh5 (Infineon):** Non-linear Vd dependency for intrinsic Impact Ionization model.
3. **2023enh6 (Infineon):** Non-linear Vd dependency for drift-region Impact Ionization model.
4. **2023enh7 (Infineon):** More accurate Vd dependency needed for expansion effect.
5. **2024enh1 (ADI):** Added additional parameters to the parameter set in nmos and pmos PARAM_Check files.
6. **2024enh2 (ADI):** Added the QA test that uses RDSMOD=1.

7. **2024enh3 (ADI):** Added the QA tests for new MULT parameters.
8. **2024enh4 (UCB/IITK):** Added new QA tests.

G. Summary of bug-fixes:

9. **2023bug10 (Infineon):** Smoothness requirement in Id-Vg for very high Vg and high Vd.
10. **2024bug1 (ADI):** Removing extra spaces reported by VAMPyRE.
11. **2024bug2 (Keysight):** Drain current discontinuity in BSIM-BULK HV model.
12. **2024bug3 (ADI):** Bug in Diode Implementation.
13. **2024bug4 (ADI):** Incorrect specification: GEOMOD and RGEOMOD as model parameters in QA tests.
14. **2024bug5 (ADI):** Updated the missing OP descriptions.
15. **2024bug6 (ADI):** Update to hypsmooth () function.
16. **2024bug7 (IITK):** Addressing potential convergence warnings by using ln_one_plus_exp () function.
17. **2024bug8 (UCB/IITK):** Update to smooth macro definition.
18. **2024bug9 (UCB/IITK):** Ensuring BETA1_i to be always non-negative number.
19. **2024bug10 (UCB/IITK):** Drain-side and Source-side drift resistance symmetry.
20. **2024bug11 (UCB/IITK):** Corrected PARAM_Check file in QA package for PMOS transistor.
21. **2024bug12 (ADI):** Instance parameters removed from model parameters list in QA package.
22. **2024bug13 (ADI):** Updated incorrect parameter description and parameter units.
23. **2024bug14 (UCB/IITK):** Removed superfluous assignments.
24. **2024bug15 (ADI):** Ensured the manual mentions the correct model name.

BSIM-BULK 107.2.0_beta3

H. Summary of enhancements:

1. **2024enh5 (ADI):** Add more QA tests to cover all parameters and MODs.

I. Summary of bug-fixes:

2. **2024bug16 (ADI):** Missing reference data in the QA package for some QA tests.
3. **2024bug17 (Infineon):** Ensure CMD1 and CMS1 parameters have impact only when IIMOD=1.

BSIM-BULK 107.2.0_beta0_1

A. 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:

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

- 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).

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

- To account for body bias dependency, ALPHA0_i is replaced with ALPHA0_eff.
- For backward compatibility, the following default values of the new parameters are used: BETA1=0, BETA2=0, BETA3=1, ALPHA1=0 and

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

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

ALPHA2=0.

- Impact Ionization (intrinsic MOS) model 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):
```

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

```
}
```

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:

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

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

- In BSIM-BULK 107.2.0_beta0_1, the VDDROP equation is modified as follows:

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

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

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

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

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

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:

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

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

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

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

Similarly,

- 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)

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

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

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
```

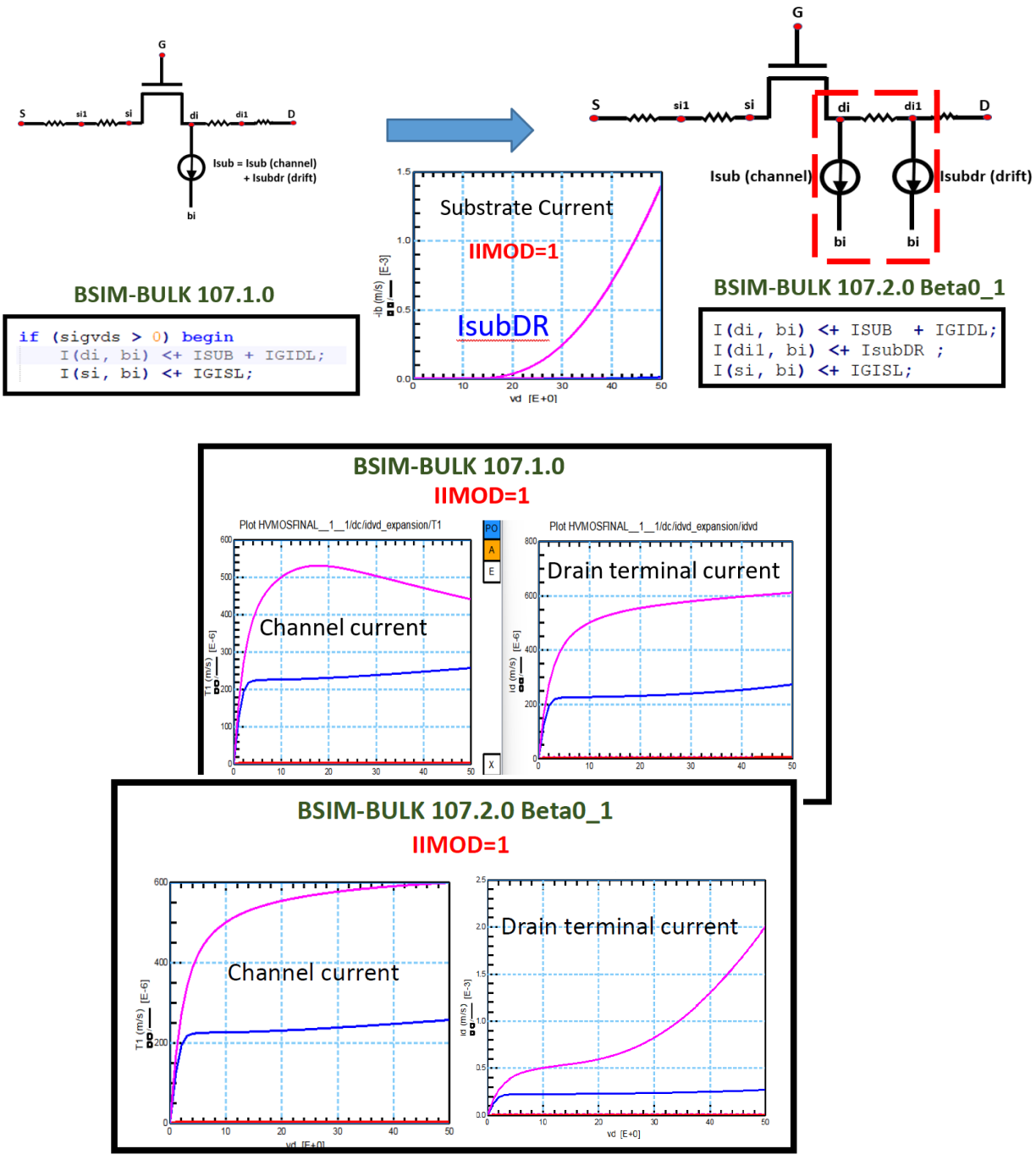
5. 2023enh1 (GF): Introduction of MULT parameters (MULT_I, MULT_Q, MULT_FN).

- 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.

B. Description of bug-fixes:

1. 2022bug9 (IITK/UCB): Correction in the 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.



2. 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.

```
3860        if (IGBMOD != 0) begin
3861            T1      = Voxm / NIGBACC_i / Vt;
3862            Vaux_Igbacc = NIGBACC_i * Vt * lln(1.0 + lexp(-T1));
3863            T2      = AIGBACC_i - BIGBACC_i * Voxmacc;
3864            T3      = 1.0 + CIGBACC_i * Voxmacc;
3865            T4      = -7.45669e11 * TOXE * T2 * T3;
3866            T5      = lexp(T4);
3867            T6      = 4.97232e-7;
3868            igbacc = NF * Weff * Leff * T6 * ToxRatio * Vg * Vaux_Igbacc * T5;
3869            igbacc = igbacc * igtemp;
3870            T1      = (Voxm - EIGBINV_i) / NIGBINV_i / Vt;
3871            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);
```

3. 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

4. 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.

```
4593    // Quantum mechanical effects
4594    `Smooth(Vt * Qb, 0.0, 0.1, qbaCV)
4595    qiaCV       = Vt * (Qs + Qd);
4596    T0          = (qiaCV + ETAQM * qbaCV) / QM0;
4597    T1          = 1.0 + pow(T0, 0.7 * BDOS);
4598    XDCinv      = ADOS * 1.9e-9 / T1;
4599    Coxeffinv   = 3.9 * `EPS0 / (BSIMBULKTOXP * 3.9 / EPSROX + XDCinv / epsratio);
4600    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

5. 2023bug1 (IITK): Typo in the bulk charge expression in manual.

From the BSIM-BULK107.1.0 manual:

Bulk charge with poly depletion effect :

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

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

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

Typo: It should be q_d

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

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

Corresponding code

```
4556    if (PCLMCV_i != 0.0) begin
4557        MdL = 1.0 + PCLMCV_i * lln(1.0 + diffVds / PCLMCV_i / Vasat);
4558    end else begin
4559        MdL = 1.0;
4560    end
4561    MdL_2       = MdL * MdL;
4562    inv_MdL     = 1.0 / MdL;
4563    inv_MdL_2   = 1.0 / MdL_2;
4564    MdL_less_1  = MdL - 1.0;
4565    vgpqm = vgfbCV - psip;
4566    DQSD  = (qs - qdeff);
4567    DQSD2 = (qs - qdeff) * (qs - qdeff);
4568    sis   = vgpqm + 2.0 * qs;
4569    sid   = vgpqm + 2.0 * qdeff;
4570    `Smooth(sis, 0.0, 0.5, T1)
4571    `Smooth(sid, 0.0, 0.5, T2)
4572    Temps = sqrt(0.25 + T1 * invgamg2);
4573    Tempd = sqrt(0.25 + T2 * invgamg2);
4574    T1 = sis / (1.0 + 2.0 * Temps);
4575    T2 = sid / (1.0 + 2.0 * Tempd);
4576    T3 = Temps + Tempd;
4577    T4 = `Oneby3 * (DQSD2 / (T3 * T3 * T3));
4578    T5 = (AbulkCV * Dvsat * inv_MdL) / (1.0 + qs + qdeff);
4579    T6 = 0.8 * (T3 * T3 + Temps * Tempd) * T5;
4580    T7 = T6 + (2.0 * invgamg2);
4581    T8 = `Oneby3 * DQSD2 * T5;
4582    dqgeff = sid * (2.0 * Tempd - 1.0) / (2.0 * Tempd + 1.0);
4583    qbeff  = vgpqm - 2.0 * (nq - 1.0) * qdeff + dqgeff;
4584    Qb  = inv_MdL * (T1 + T2 + (T4 * T7 - nq * (qs + qdeff + T8))) + MdL_less_1 * qbeff;
```

- 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 (See Appendix B).

6. 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

7. 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:

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

- In BSIM-BULK107.2.0_beta0_1, the expression for E_m is corrected as follows:

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

BSIM-BULK 107.2.0_beta0_2

C. Description of enhancements:

1. 2023enh2 (GF): Flicker noise model enhancement.

- In BSIM-BULK107.1.0, the flicker noise model is not able to capture the drain current noise spectral density (S_{id}) data in the weak inversion.
- Flicker noise model in weak inversion in BSIM-BULK107.1.0 is:

$$S_{wi} = \frac{\boldsymbol{NOIA}.kT.I_{ds}^2}{W_{eff}.NF.L_{eff,noi}.10^{10}.N^{*2}}$$

- In BSIM-BULK107.2.0_beta0_2, a bias dependent effective NOIA is introduced to model the noise behavior in weak inversion:

$$\boldsymbol{NOIA_{eff}} = Max\left(1, \left(\frac{\frac{NOIA3}{NOIA}}{1+\left(\frac{q_{ia}}{QSREF}\right)^{MPOWER}}\right)\right) * \boldsymbol{NOIA}$$

$$S_{wi} = \frac{\boldsymbol{NOIA_{eff}}.kT.I_{ds}^2}{W_{eff}.NF.L_{eff,noi}.10^{10}.N^{*2}}$$

- Following is the Verilog-A implementation of bias dependent $\boldsymbol{NOIA_{eff}}$ in the BSIM-BULK107.2.0_beta0_2:

```
if (NOIA > 0.0 || NOIB > 0.0 || NOIC > 0.0) begin
    if (NOIA3 != 0 && NOIA > 0) begin
        T1 = qia/QSREF_i;
        T2 = 1 + pow(T1, MPOWER_i);
        T3 = NOIA3_i/T2;
        T4 = T3 / NOIA;
        T5 = 0.5 * (T4 + 1 + sqrt((T4 - 1) * (T4 - 1) + 0.25 * SPFN * SPFN));
        NOIAeff = NOIA * T5;
    end else begin
        NOIAeff = NOIA;
    end
```

- The default value of NOIA3 is 0 making the enhancement backward compatible.

- New parameters introduced in BSIM-BULK107.2.0_beta0_2 for this enhancement: **NOIA3**, **MPOWER**, **QSREF**, **SPFN**.

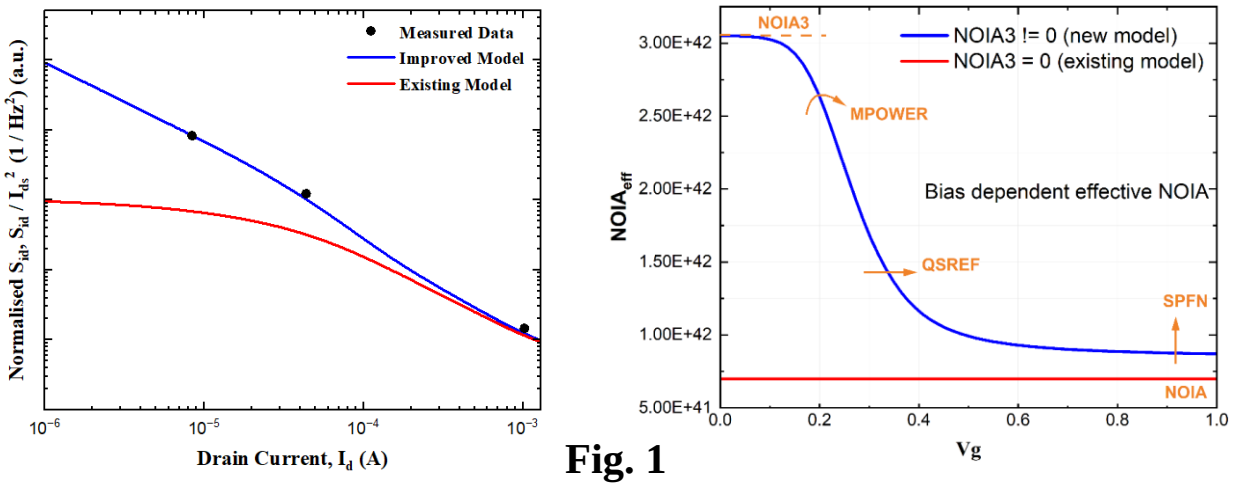


Fig. 1

Fig. 1: A bias dependent NOIA is introduced when NOIA3 > 0 to model noise behavior in weak inversion region. The new model is able to match the Sid/Id^2 trend vs current data in weak inversion region which existing model failed to capture.

2. 2023enh3 (GF): Decoupling drain-side drift resistance for capacitance calculations.

- We have decoupled drain-side drift resistance used in I-V and C- V calculations with a new parameter RDLCWCV used in capacitance calculations.
- The default value of RDLCWCV is RDLCW and therefore, for the default value, the drain-side drift resistance used in current and capacitance calculations will be the same.
- Decoupling is done in such a way that the internal drain potential, V(di), used in capacitance calculations is modified as shown below.

$$V(di)_{CV} = V(di)_{IV} + devsign * (1.0 - \frac{RDLCWCV}{RDLCW}) * V(di1, di)$$

- This new enhancement provides more flexibility in capacitance fitting without affecting the current characteristic fitting.

- An example case is shown below. Fig. 2 shows the capacitance plots using BSIM-BULK 107.2.0_beta0_1 and Fig. 3 shows the same using the new BSIM-BULK 107.2.0_beta0_2 code.

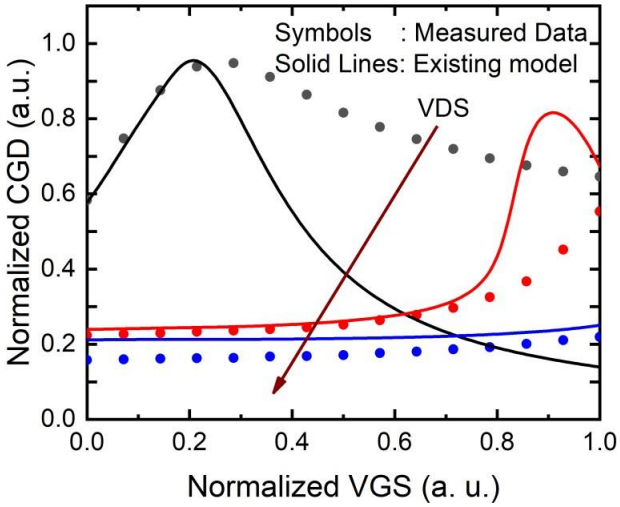


Fig. 2: Capacitance plots obtained using BSIM-BULK 107.2.0_beta0_1 code.

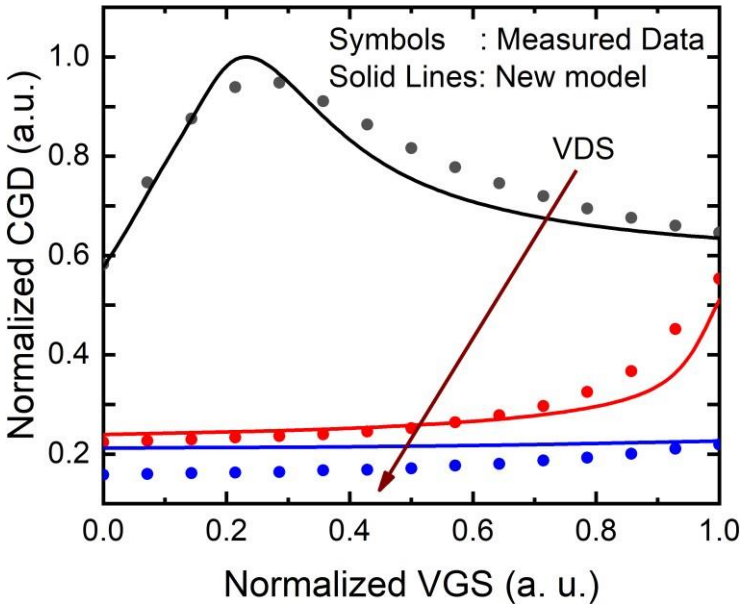


Fig. 3: Capacitance plots obtained using BSIM-BULK 107.2.0beta0_2 code.

D. Description of bug-fixes:

1. 2023bug4 (ADI): Issue in C_{GD} reciprocity while exercising QM parameters.

- C_{GD} non-reciprocity was observed at 0V V_d bias while using QM parameters.

Semi Physical expression [1]

The inversion charge layer thickness is given by

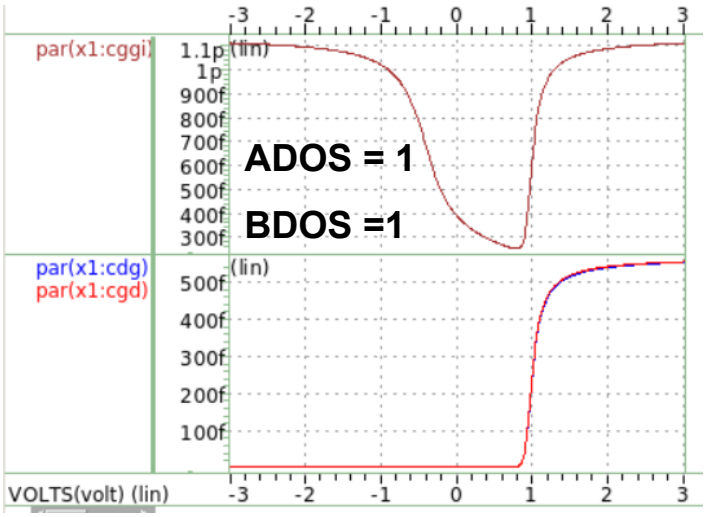
$$X_{DC} = \frac{\beta}{\alpha + \left(\frac{V_{gtx} + 4(V_T - V_{fb} - \varphi_{s0})}{2T_{ox}}\right)^{0.7}} \quad (5)$$

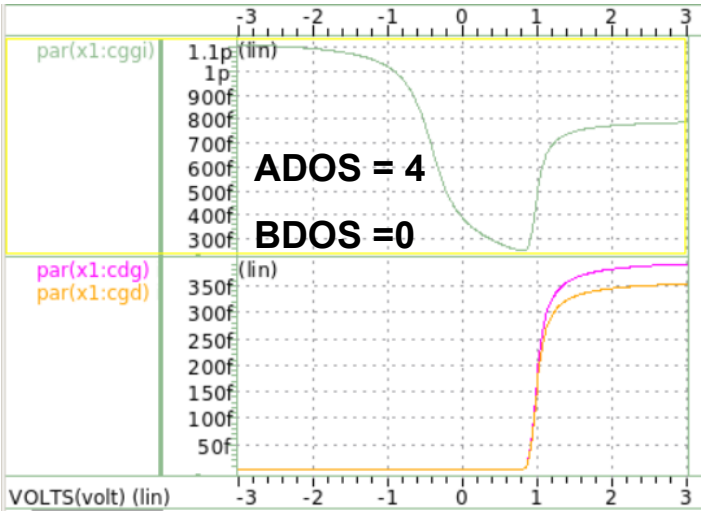
where $\alpha = 1\ (\mathrm{MV/cm})^{0.7}$, $\beta = 1.9 \times 10^{-7}\ (\mathrm{cm(MV/cm)}^{0.7})$, and

Parameterized expression in BSIM-BULK

$$X_{DC}^{inv} = \frac{ADOS \cdot (1.9 \cdot 10^{-9})}{1 + \left[\frac{Q_i + ETAQM \cdot Q_B}{QM0}\right]^{0.7*BDOS}}$$

- We investigated the issue and concluded that non-reciprocity issue arises when the parameterized equation deviates too much from semi-physical expression (i.e., when BDOS=0, or based on parameter selection, a negligible contribution of the second term in the denominator).





- Also, physically [1,2] the QM contribution to capacitance typically runs between 10-15%, and we recommend not using the extreme values of QM parameters to achieve it. Furthermore, we have not restricted the parameter values for good fitting flexibility.

- We, however, have added a suggestion note in the Technical Manual for the same:
  "QM contribution to capacitance is usually between 10-15%, and we recommend not using extreme values of QM parameters to achieve it".

[1] W. Liu, X. Jin, Y. King and C. Hu, TED, May 1999.

[2] Gildenblat, G., 2010. Compact modeling (p. 4). Netherlands: Springer.

2. 2023bug5 (ADI): MULT implementation correction.

- The VAMPyRE v1.9.2 reported some errors (missing/extra contributions) in the BSIM-BULK 107.2.0_beta0_1.
- The reported errors were addressed in BSIM-BULK 107.2.0_beta0_2 as follows:

```
ISUB = Iii * MULT_I * devsign;
```

```
I(di, si) <+ devsign * MULT_I * Issl;
```

```
ISUB = Iii * MULT_I * devsign;
```

```
I(N2)   <+   white_noise(MULT_I * cm_igid, "corl");
I(NI)   <+   white_noise(MULT_I * sqig * sqig * (1.0 - cm_igid), "corl");
I(NI)   <+   -sqig * MULT_I * V(N2);
I(NC)   <+   MULT_Q * ddt(mig * Cox * Weff * NF * Leff * V(NC));
```

```
I(N2) <+ MULT_I * V(N2);
I(NR) <+ MULT_I * V(NR);
```

```
I(di, bi) <+ MULT_I * ISUB + IGIDL;
```

```
I(si, bi) <+ MULT_I * ISUB + IGISL;
```

```
I(gm, gi) <+ V(gm, gi) * MULT_I * Gcrg;
```

MISSING CONTRIBUTION (in BSIM-BULK 107.2.0 Beta0_1)
THAT IS ADDED IN BSIM-BULK 107.2.0 Beta0_2

EXTRA CONTRIBUTION (in BSIM-BULK 107.2.0 Beta0_1)
THAT IS REMOVED IN BSIM-BULK 107.2.0 Beta0_2

3. 2023bug6 (ADI): Correction in units of binning parameters.

- The VAMPyRE v1.9.2 reported some inconsistency in units of some binning parameters in the BSIM-BULK 107.2.0_beta0_1.

- For example, VAMPyRE reported units of LK2, WK2 and PK2 were incorrect.

K2: Vth shift due to vertical non-uniform doping.

K2_i = K2 + BIN_L * LK2 + BIN_W * WK2 + BIN_WL * PK2;

(Binning equation for K2)

- K2_i and K2 have units of V.
- Units of BIN_L, BIN_W and BIN_WL are m^{-1}, m^{-1} and m^{-2} respectively.
- In BSIM-BULK 107.2.0_beta0_1, Units of LK2, WK2 and PK2 were m, m and m^2 respectively.
- To maintain dimensional consistency, the binning parameters (LK2, WK2 and PK2) in BSIM-BULK 107.2.0_beta0_2 were corrected to V*m, V*m and V*m^2 respectively.
- Binning parameters were corrected in BSIM-BULK107.2.0_beta0_2 for 15 following parameters:

  K2, RSWMIN, RSW, RDWMIN, RDW, RDSWMIN, RDSW, FPROUT, AGIDL, AGISL, CGSL, K2EDGE, KVTH0EDGE, KVTH0EDGEWE, K2EDGEWE.

4. 2023bug7 (ADI): Removing superfluous assignment of variables.

- The VAMPyRE v1.9.2 reported superfluous assignment of some variables in the BSIM-BULK 107.2.0_beta0_1.

  Following is a list of assignments/initialization that were dropped in BSIM-BULK107.2.0_beta0_2:

1. `local_sca = 0.0;` **(initialized twice)**
2. `local_scb = 0.0;` **(initialized twice)**
3. `T9 variable assignment statement in BSIM_q macro` **(superfluous assignment)**
4. `Czbdswg = 0.0;` `(no gate-edge contribution to Qbdj_ext)` **(superfluous initialization)**
5. `gamg2 variable in CVMOD = 0` **(superfluous assignment)**
6. `sqrtPhist variable assignment in EDGEFET == 1` **(superfluous assignment)**
7. `inv_gam variable assignment statement in EDGEFET == 1` **(superfluous assignment)**

5. 2023bug8 (ADI): Addressing the bias dependent $strobe warnings reported by VAMPyRE.

- The VAMPyRE v1.9.2 reported some bias-dependent $strobe conditions in the BSIM-BULK107.2.0_beta0_1.
  -

```
WARNING in file before.va, line 3032: bias-dependent $strobe() may degrade performance
```

```
VSAT_t  = VSAT_i * pow(TRatio, -AT_i);
if (VSAT_t < 100.0) begin
    $strobe("Warning: VSAT(%f) = %e is less than 100, setting it to 100.", DevTemp, VSAT t);
    VSAT_t = 100.0;
end
```

BSIMBULK107.2.0_beta0_1

The $strobe statement was removed in this case as follows:

```
VSAT_t  = VSAT_i * pow(TRatio, -AT_i);
if (VSAT_t < 100.0) begin
    VSAT_t = 100.0;
end
```

BSIMBULK107.2.0_beta0_2

  -

```
WARNING in file before.va, line 3042: bias-dependent $strobe() may degrade performance
```

```
VSATR_t = VSATR_i * pow(TRatio, -AT_i);
if (VSATR_t < 100.0) begin
    $strobe("Warning: VSATR(%f) = %e is less than 100, setting it to 100.", DevTemp, VSATR_t);
    VSATR_t = 100.0;
end
```

BSIMBULK107.2.0_beta0_1

The $strobe statement was removed in this case as follows:

```
VSATR_t = VSATR_i * pow(TRatio, -AT_i);
if (VSATR_t < 100.0) begin
    VSATR_t = 100.0;
end
```

BSIMBULK107.2.0_beta0_2

  -

```
WARNING in file before.va, line 3048: bias-dependent $strobe() may degrade performance
```

```
VSATCV_t = VSATCV_i * pow(TRatio, -AT_i);
if (VSATCV_t < 100.0) begin
    $strobe("Warning: VSATCV(%f) = %e is less than 100, setting it to 100.", DevTemp, VSATCV_t);
    VSATCV_t = 100.0;
end
```

BSIMBULK107.2.0_beta0_1

The $strobe statement was removed in this case as follows:

```
VSATCV_t = VSATCV_i * pow(TRatio, -AT_i);
if (VSATCV_t < 100.0) begin
    VSATCV_t = 100.0;
end
```

BSIMBULK107.2.0_beta0_2

-

```
WARNING in file before.va, line 3117: bias-dependent $strobe() may degrade performance
WARNING in file before.va, line 3126: bias-dependent $strobe() may degrade performance
WARNING in file before.va, line 3140: bias-dependent $strobe() may degrade performance
WARNING in file before.va, line 3155: bias-dependent $strobe() may degrade performance
```

These warnings were removed by modifying **`BSIMBULKPAeffGeo`** macro as follows:

```
`define BSIMBULKPAeffGeo(nf, geo, minSD,Weffcj, DMCG, DMCI, DMDG, Ps, Pd, As, Ad) \
    begin if (geo < 9) \
        `BSIMBULKNumFingerDiff(nf, minSD, nuIntD, nuEndD, nuIntS, nuEndS) \
        T0 = DMCG + DMCI; \
        T1 = DMCG + DMCG; \
        T2 = DMDG + DMDG; \
        PSiso = T0 + T0 + Weffcj; \
```

BSIMBULK107.2.0_beta0_1

```
`define BSIMBULKPAeffGeo(nf, geo, minSD,Weffcj, DMCG, DMCI, DMDG, Ps, Pd, As, Ad) \
    begin \
        if (geo < 9) begin \
            `BSIMBULKNumFingerDiff(nf, minSD, nuIntD, nuEndD, nuIntS, nuEndS) \
        end \
        T0 = DMCG + DMCI; \
        T1 = DMCG + DMCG; \
        T2 = DMDG + DMDG; \
```

BSIMBULK107.2.0_beta0_2

-

```
WARNING in file before.va, line 4371: bias-dependent $strobe() may degrade performance
```

The below highlighted portion in Halo flicker noise calculation was moved to Parameter checking section.

```
if (FNOIMOD == 1) begin
    LH1 = LH;
    if (Leff > LH1) begin
        T0 = (Leff - LH1);
    end else begin
        LH1 = Leff;
        T0 = LH1;
    end
    if (LINTNOI >= T0 / 2.0) begin
        $strobe("Warning: LINTNOI = %e is too large - Leff for noise is negative. Re-setting LINTNOI = 0.", LINTNOI);
        LINTNOI_i = 0.0;
    end else begin
        LINTNOI_i = LINTNOI;
    end
    LeffnoiH = Leff;
    vgfbh  = (Vg - VFB_i) / Vt;
    gam_h  = sqrt(2.0 * `q * epssi * HNDEP / Vt) / Cox;
    phib_h = ln(HNDEP / ni);
```

BSIMBULK107.2.0_beta0_1

```
if (FNOIMOD == 1) begin
    LH1 = LH;
    if (Leff > LH1) begin
        T0 = (Leff - LH1);
    end else begin
        LH1 = Leff;
        T0 = LH1;
    end
    if (LINTNOI >= T0 / 2.0) begin
        $strobe("Warning: LINTNOI = %e is too large - Leff for noise is negative. Re-setting LINTNOI = 0.", LINTNOI);
        LINTNOI_i = 0.0;
    end else begin
        LINTNOI_i = LINTNOI;
    end
end
```

Parameter checking section in BSIMBULK107.2.0_beta0_2

BSIM-BULK 107.2.0_beta1

E. Description of bug-fixes:

1. 2023bug8 (ADI): Removing the redundant 0V source in TNOIMOD=0.

- In BSIM-BULK107.2.0_beta0_2, internal node N1 is tied to ground with a 0V voltage source as shown below:

```
4546        case (TNOIMOD)
4547            0: begin
4548                QSi   = -NF * Weff * Leff * Cox * Vt * Qs;
4549                QDi   = -NF * Weff * Leff * Cox * Vt * Qd;
4550                T0    = ueff * abs(QSi + QDi);
4551                T1    = T0 * Rdsi + Leff * Leff;
4552                Gtnoi = (T0 / T1) * NTNOI;
4553                sidn  = Nt * Gtnoi;
4554                I(di, si) <+ white_noise(MULT_I * sidn, "id");
4555                V(N1)     <+ 0.0;
4556            end
```

BSIM-BULK107.2.0_beta0_2

- Outside the “case (TNOIMOD)” we have following highlighted lines

which provide 1-ohm resistors in order to ground the two internal nodes N2 and N1 due to definition “branch (N1) NR;”

```
4546        case (TNOIMOD)
4547            0: begin
4557            1: begin
4597        endcase
4598        I(N2) <+ V(N2);
4599        I(NR) <+ V(NR);
```

- Since we already have a resistor tying N1 to ground, and no other sources on N1 in TNOIMOD=0, there is no need for adding explicit 0V source.
- In some simulators, 0V source adds an extra row to the circuit matrix, for the current through the source. Therefore, in BSIM-BULK107.2.0 Beta 1, the line “V(N1) <+ 0.0;” is removed in TNOIMOD=0.

```
4546        case (TNOIMOD)
4547            0: begin
4548                QSi   = -NF * Weff * Leff * Cox * Vt * Qs;
4549                QDi   = -NF * Weff * Leff * Cox * Vt * Qd;
4550                T0    = ueff * abs(QSi + QDi);
4551                T1    = T0 * Rdsi + Leff * Leff;
4552                Gtnoi = (T0 / T1) * NTNOI;
4553                sidn  = Nt * Gtnoi;
4554                I(di, si) <+ white_noise(MULT_I * sidn, "id");
4555            end
4556            1: begin
4596        endcase
4597        I(N2) <+ V(N2);
4598        I(NR) <+ V(NR);
```

BSIM-BULK107.2.0_beta1

BSIM-BULK 107.2.0_beta2

F. Description of enhancements:

1. 2023enh4 (ADI): Fitting flexibility of Cgg in moderate inversion.

- The equation solved for computing the normalized inversion charge, q_i, in BSIM-BULK 107.2.0 beta1 is shown below.

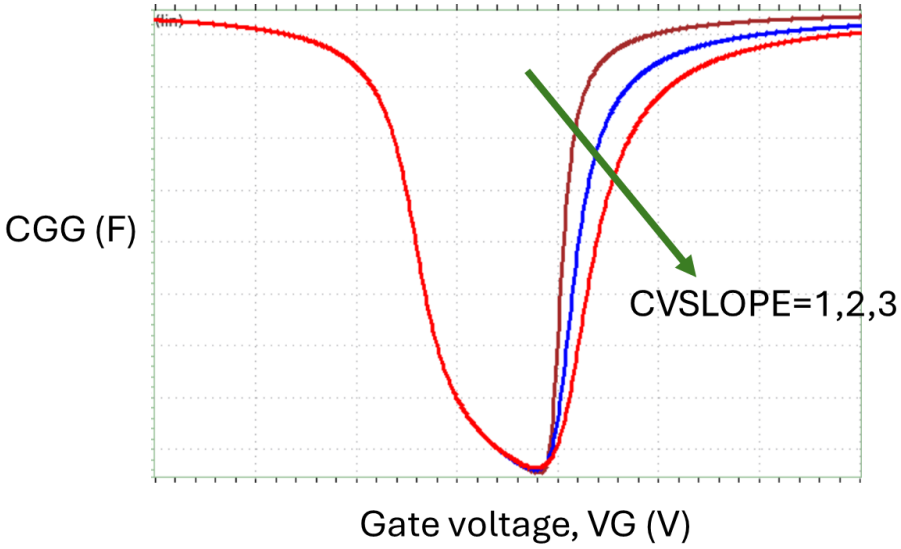
$$\ln\left[\frac{2n_q q_i}{\gamma_0}\left\{\frac{2n_q q_i}{\gamma_0}+2\sqrt{\psi_p-2q_i}\right\}\right]+2q_i=\psi_p-2\phi_f-V_{ch}$$

- In BSIM-BULK 107.2.0 beta2, for CVMOD=1, the equation solved has been

modified as shown below.

$$\boldsymbol{CVSLOPE}\ln\left[\frac{2n_q q_i}{\gamma_0}\left\{\frac{2n_q q_i}{\gamma_0}+2\sqrt{\psi_p-2q_i}\right\}\right]+2q_i=\psi_p-2\phi_f-V_{ch}$$

- CVSLOPE is a new parameter that can help in tuning the slope of the CGG capacitance v/s gate voltage plot in depletion to strong inversion transition region.
- The plots of CGG v/s gate voltage for different CVSLOPE values are shown below.



- Default value of CVSLOPE=1, which ensures backward compatibility of the code.

2. 2023enh5 (Infineon): Non-linear Vd dependency for Intrinsic Impact Ionization model.

- BSIM-BULK107.2.0 beta1 was not able to capture the nonlinear Vd dependence accurately in Ib-Vg characteristics.

- In BSIM-BULK107.2.0 beta2, we introduce two additional parameters (**ALPHA3** and **ALPHA4**) to capture this non-linear dependence accurately.
- Following is the intrinsic impact ionization model in BSIM-BULK107.2.0 beta1:

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

$$diffVdsii = V_{ds} - Vdseffii$$

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

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

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

- In BSIM-BULK107.2.0 beta2 $ALPHA0\_eff$ was modified as follows:

$$ALPHA\_eff = \frac{\boldsymbol{ALPHA0}}{T1} * (1 + ALPHA1 * Vbsx + ALPHA2 * Vbsx^2)$$

$$T1 = (1 + \boldsymbol{ALPHA4} * exp(\boldsymbol{ALPHA3} * Vdsx))$$

- The default values of new parameters are: ALPHA3=0 and ALPHA4=0. Thus, this change is backward compatible.
- We introduced geometry width scaling parameters (**ALPHA0W** and **ALPHA0WEXP**) for ALPHA0 in addition to already existing geometry length scaling parameters (**ALPHA0L** and **ALPHA0LEXP**).
- Also, geometry width scaling parameters for width were introduced in BETA0, BETA1 and BETA2, to enhance the model flexibility.
- Following are the new geometry scaling parameters introduced in BSIM-BULK 107.2.0 beta2: **ALPHA0W**, **ALPHA0WEXP**, **BETA0W**, **BETA0WEXP**,

BETA1W, **BETA1WEXP**, **BETA2W**, **BETA2WEXP**.

3. 2023enh6 (Infineon): Non-linear Vd dependency for drift-region Impact Ionization model.

- BSIM-BULK107.2.0 beta1 was not able to accurately capture the non-linear Vd dependence in Ib-Vg characteristics and Vg dependence in Ib-Vd characteristics simultaneously.
- In BSIM-BULK107.2.0 beta2, we have introduced a decoupled vdrift calculation by introducing three additional parameters **PTWGHVII**, **PTWGHV1II** and **PSATXHVII**.
- Also, in BSIM-BULK 107.2.0 beta1 $ALPHADR\_eff$ and VDDROP were given by:

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

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

- In BSIM-BULK 107.2.0 beta2 $ALPHADR\_eff$ was modified as:

$$T4 = \boldsymbol{ALPHADR1} * Vbsx + \boldsymbol{ALPHADR2} * Vbsx^2$$

$$T5 = \boldsymbol{ALPHADR3} * VDDROP + \boldsymbol{ALPHADR4} * VDDROP^{\boldsymbol{DREXP}}$$

$$ALPHADR\_eff = \boldsymbol{ALPHADR} * (1 + T4 + T5)$$

- The default values of new parameters are: ALPHADR3=0, ALPHADR4=0, DREXP=1, PTWGHVII=0, PTWGHV1II=0 and PSATXHVII=60. Thus, making this change backward compatible.

4. 2023enh7 (Infineon): More accurate Vd dependency needed for expansion effect.

- In BSIM-BULK107.2.0 beta1, there was not enough flexibility in capturing capture the Ib-Vd and Id-Vd measured data.

- In BSIM-BULK107.2.0 beta2, the drift region II model was enhanced, resulting in accurate capture of Vd dependency, especially for the initial signature of expansion effect in the measured data.

5. 2024enh1 (ADI): Added additional parameters to the parameter set in nmos and pmos PARAM_Check files.

- nmosPARAM_Check and pmosPARAM_Check files are supposed to test all the parameters set to their default values.
- In BSIM-BULK107.2.0 beta 1, 24 parameters were not present in the parameter set of the above two files.
- In BSIM-BULK107.2.0 beta 2, including the parameters newly added, the following 34 additional parameters were added to the nmosPARAM_Check and pmosPARAM_Check files:
  MULT_I, MULT_Q, MULT_FN, ALPHA1, ALPHA2, ALPHADR1, ALPHADR2, DRII3, DRII4, CMD1, CMD2, CMS1, CMS2, BETA1, BETA2, BETA3, ALPHA0R, BETA0R, SPQBACV, NOIA3, MPOWER, QSREF, SPFN, RDLCWCV, ALPHA3, ALPHA4, ALPHADR3, ALPHADR4, DREXP, PTWGHVII, PTWGHV1II, PSATXHVII, DSMOOTH and CVSLOPE.
- These were set to default values.

6. 2024enh2 (ADI): Added the QA test that uses RDSMOD=1.

- In BSIM-BULK107.2.0 beta 1, there was no QA test using RDSMOD=1.
- In BSIM-BULK107.2.0 beta 2, QA tests 48 to 66 are added (in nmos and pmos qaSpec files).
- QA tests 48 to 66 in BSIM-BULK107.2.0 beta 2 are same as the QA tests 1 to 19 except that RDSMOD=1.

6. 2024enh3 (ADI): Added the QA test for new MULT parameters.

- In BSIM-BULK107.2.0_beta2, following three QA tests were added (in nmos and pmos qaSpec files) for the new MULT parameters (MULT_I, MULT_Q, MULT_FN):

added to qaSpec file (nmos)

```
434  //MULT_I QA: MULT_I=2
435  test                  046_DC_multi
436  biases                V(s)=0 V(b)=0
437  biasList              V(g)=0.15,0.5,0.75,1.0
438  biasSweep             V(d)=0,1.2,0.1
439  outputs               I(d)
440  instanceParameters    W=10.0e-6 L=1e-6 MULT_I=2
441  modelParameters       parameters/nmosParameters
```

added to qaSpec file (pmos)

```
434  //MULT_I QA: MULT_I=2
435  test                  046_DC_multi
436  biases                V(s)=0 V(b)=0
437  biasList              V(g)=-0.15,-0.5,-0.75,-1.0
438  biasSweep             V(d)=0,-1.2,-0.1
439  outputs               I(d)
440  instanceParameters    W=10.0e-6 L=1e-6 MULT_I=2
441  modelParameters       parameters/pmosParameters
```

added to qaSpec file (nmos)

```
443  //MULT_Q QA: MULT_Q=2
444  test                  047_AC_multq
445  biases                V(s)=0 V(b)=0
446  biasList              V(d)=0.1,1.0
447  biasSweep             V(g)=-1.2,1.2,0.2
448  outputs               C(g,g) C(g,d) C(g,s)
449  instanceParameters    W=10.0e-6 L=1e-6 MULT_Q=2
450  modelParameters       parameters/nmosParameters
```

added to qaSpec file (pmos)

```
443  //MULT_Q QA: MULT_Q=2
444  test                  047_AC_multq
445  biases                V(s)=0 V(b)=0
446  biasList              V(d)=-0.1,-1.0
447  biasSweep             V(g)=-1.2,1.2,0.2
448  outputs               C(g,g) C(g,d) C(g,s)
449  instanceParameters    W=10.0e-6 L=1e-6 MULT_Q=2
450  modelParameters       parameters/pmosParameters
```

added to qaSpec file (nmos)

```
452  // MULT_FN QA: MULT_FN=2
453  test                  048_Noise_multfn
454  biases                V(s)=0 V(b)=0 V(d)=1.0
455  biasList              V(g)=0.6,0.8,1.0
456  freq                  dec 1 1e2 1e10
457  outputs               N(d)
458  instanceParameters    W=10.0e-3 L=0.1e-6 MULT_FN=2
459  modelParameters       parameters/nmosParameters
```

added to qaSpec file (pmos)

```
452  // MULT_FN QA: MULT_FN=2
453  test                  048_Noise_multfn
454  biases                V(s)=0 V(b)=0 V(d)=-1.0
455  biasList              V(g)=-0.6,-0.8,-1.0
456  freq                  dec 1 1e2 1e10
457  outputs               N(d)
458  instanceParameters    W=10.0e-3 L=0.1e-6 MULT_FN=2
459  modelParameters       parameters/pmosParameters
```

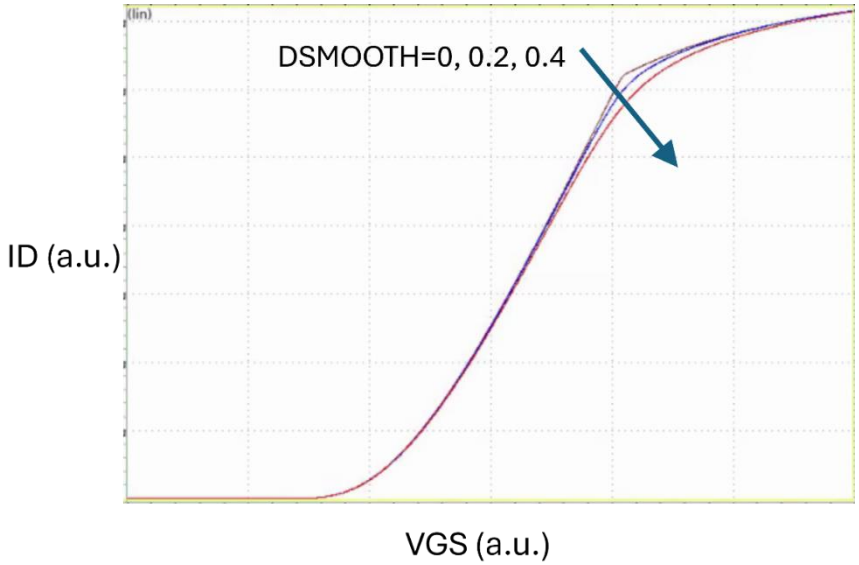
8. 2024enh4 (UCB/IITK): Added new QA tests.

- Added QA tests for RDSMOD=0, HVMOD=1, HVCAP=1, HVCAPS=1, RGATEMOD=1, RGATEMOD=2, RBODYMOD=1, RBODYMOD=2, RBODYHVMOD=1, CVMOD=1, COVMOD=0, COVMOD=1 and SHMOD=1.
- Total number of QA tests have increased from 44 in BSIM-BULK 107.2.0 beta1 to 133 in BSIM-BULK 107.2.0 beta2.

G. Description of bug-fixes:

9. 2023bug10 (Infineon): Smoothness requirement in Id-Vg for very high Vg and high Vd

- For HV devices, with large drift region, the smoothness with which the drain current starts to saturate when Vg increases cannot be controlled in BSIM-BULK 107.2.0 beta1 code.
- In BSIM-BULK 107.2.0 beta2 code, we have introduced a new parameter DSMOOTH which will smoothly control the transition of the channel current ids to the drift saturation current.
- Sample plot is shown below.



- The new code segment of BSIM-BULK 107.2.0 beta2 is shown below.

```
if (RDLCW != 0 && RSLCW != 0) begin
    T5 = sigvds * ids / min(idrift_sat_d, idrift_sat_s);
    `Smooth2(T5, 1.0, DSMOOTH, T5)
    T5 = T5 + 0.5 * sqrt(1.0 * 1.0 + 0.25 * DSMOOTH * DSMOOTH) - 0.5 - 0.25 * DSMOOTH;
    `Smooth(T5, -1.0, DSMOOTH, T5)
    T5 = T5 - 0.5 * sqrt(1.0 * 1.0 + 0.25 * DSMOOTH * DSMOOTH) + 0.5;
    ids = sigvds * min(idrift_sat_d, idrift_sat_s) * T5;
end else begin
    if (RDLCW != 0) begin
        T5 = sigvds * ids / idrift_sat_d;
        `Smooth2(T5, 1.0, DSMOOTH, T5)
        T5 = T5 + 0.5 * sqrt(1.0 * 1.0 + 0.25 * DSMOOTH * DSMOOTH) - 0.5 - 0.25 * DSMOOTH;
        `Smooth(T5, -1.0, DSMOOTH, T5)
        T5 = T5 - 0.5 * sqrt(1.0 * 1.0 + 0.25 * DSMOOTH * DSMOOTH) + 0.5;
        ids = sigvds * idrift_sat_d * T5;
    end
    if (RSLCW != 0) begin
        T5 = sigvds * ids / idrift_sat_s;
        `Smooth2(T5, 1.0, DSMOOTH, T5)
        T5 = T5 + 0.5 * sqrt(1.0 * 1.0 + 0.25 * DSMOOTH * DSMOOTH) - 0.5 - 0.25 * DSMOOTH;
        `Smooth(T5, -1.0, DSMOOTH, T5)
        T5 = T5 - 0.5 * sqrt(1.0 * 1.0 + 0.25 * DSMOOTH * DSMOOTH) + 0.5;
        ids = sigvds * idrift_sat_s * T5;
    end
end
```

- This is not a backward compatible change. However, backward incompatibility is only for HV devices (HVMOD=1).

10. 2024bug1 (ADI): Removing the extra spaces reported by VAMPyRE.

- We removed extra spaces reported by VAMPyRE near the end of following lines in BSIM-BULK 107.2.0 beta1 code:

```
743— `MPRnb( PNDEP             ,0.0                                    ,"1/m"                                  ,"Area dependence of NDEP " )
1432— `MPRcz( XRCRG1           ,12.0                                   ,""                                     ,"1st fitting parameter the bias-dependent Rg " )
1433— `MPRcz( XRCRG2           ,1.0                                    ,""                                     ,"2nd fitting parameter the bias-dependent Rg " )
1826— `MPRoz( SLHV1            ,1.0                ,""                                 ,"Parameter for slope of the accumulation capacitance " )
```

11. 2024bug2 (Keysight): Drain current discontinuity in BSIM-BULK HV model.

- Low value of MDRIFT can cause numerical round-off issue resulting in

discontinuity in the drain current.

- In BSIM-BULK 107.2.0 beta1 code, the value of MDRIFT can be any number between 0 and 4.
- In BSIM-BULK 107.2.0 beta2 code, the macro definition of MDRIFT has been modified to ensure that MDRIFT cannot be less than 0.5.

12. 2024bug3 (ADI): Bug in Diode Implementation

- A code snippet of BSIM-BULK 107.2.0 beta1 implementation for RBODYMOD≠0 and RBODYHVMOD=1 is shown below.

  BSIM-BULK 107.2.0 beta1

```
if (RBODYMOD != 0 && RBODYHVMOD == 1) begin
    I(ddbulk, d) <+ V(ddbulk, d) * MULT_I * Grdb;
    I(ddbulk, d) <+ white_noise(Nt * MULT_I * Grdb, "rdb");
end else begin
    V(d, ddbulk)  <+ 0.0;
end

// Diode currents and capacitances HV
if (RBODYMOD != 0 && RBODYHVMOD == 1) begin
    I(dbulk, di) <+ devsign * MULT_I * Ibd + (1.0 - XPART) * MULT_I * V(dbulk, di) * gmin;
    I(dbulk, ddbulk) <+ devsign * MULT_I * Ibd_ext + XPART * MULT_I * V(dbulk, ddbulk) * gmin;
    I(dbulk, di) <+ devsign * ddt(MULT_Q * Qbdj);
    I(dbulk, ddbulk) <+ devsign * ddt(MULT_Q * Qbdj_ext);
end else begin
    V(d, ddbulk)  <+ 0.0;
end
```

Same contribution repeated twice

Same contribution repeated twice

- As it can be seen from the above implementation, the else condition in the diode implementation is unnecessary as it has already been taken care of in the first if-else block.
- Therefore, the implementation in BSIM-BULK 107.2.0 beta2 has been modified as shown below.
  - BSIM-BULK 107.2.0 beta2

```
if (RBODYMOD != 0 && RBODYHVMOD == 1) begin
    I(ddbulk, d) <+ V(ddbulk, d) * MULT_I * Grdb;
    I(ddbulk, d) <+ white_noise(Nt * MULT_I * Grdb, "rdb");
end else begin
    V(d, ddbulk)  <+ 0.0;
end

// Diode currents and capacitances HV
if (RBODYMOD != 0 && RBODYHVMOD == 1) begin
    I(dbulk, di) <+ devsign * MULT_I * Ibd + (1.0 - XPART) * MULT_I * V(dbulk, di) * gmin;
    I(dbulk, ddbulk) <+ devsign * MULT_I * Ibd_ext + XPART * MULT_I * V(dbulk, ddbulk) * gmin;
    I(dbulk, di) <+ devsign * ddt(MULT_Q * Qbdj);
    I(dbulk, ddbulk) <+ devsign * ddt(MULT_Q * Qbdj_ext);
end
```

13. 2024bug4 (ADI): Incorrect specification: GEOMOD and RGEOMOD as model parameters in QA tests.

- In the qaSpec file of the QA package of BSIM-BULK 107.2.0 beta1, the parameters RGEOMOD and GEOMOD were included as model parameters.
- RGEOMOD and GEOMOD are instance parameters.
- In the QA package of BSIM-BULK 107.2.0 beta2, the parameters RGEOMOD and GEOMOD are removed from model parameters' list and are correctly included as instance parameters.

14. 2024bug5 (ADI): Updated the missing/incorrect OP descriptions.

- Description of some of the operating points was missing in the BSIM-BULK 107.2.0 beta1 code.
- In the revised BSIM-BULK 107.2.0 beta2 code, the description for those operating points has been added.
- Description of the operating point CGBOV has been correctly updated.

BSIM-BULK 107.2.0 beta1

```
`OPM( CGBOV,        "F",     "Front gate charge" )
```

BSIM-BULK 107.2.0 beta2

```
`OPM( CGBOV,  "F",  "Gate-to-substrate overlap capacitance" )
```

15. 2024bug6 (ADI): Update to hypsmooth () function.

- BSIM-BULK 107.2.0 beta1 uses hypsmooth (x, c) to prevent division by zero.
- When |x| >> c, numerical problems (roundoff) can occur. Because in this case (x * x + 4.0 * c * c) is numerically equal to |x|.
- Therefore, hypsmooth (x, c) is numerically exactly 0 for large negative x.
- Although extra terms and offsets are used in many places in the code to prevent this:

```
// All NJT*'s smoothed to 0.01 to prevent divide by zero/negative values
NJTS_t     = hypsmooth(NJTS * (1.0 + TNJTS * (TRatio - 1.0)) - 0.01, 1.0e-3) + 0.01;
NJTSSW_t   = hypsmooth(NJTSSW * (1.0 + TNJTSSW * (TRatio - 1.0)) - 0.01, 1.0e-3) + 0.01;
NJTSSWG_t  = hypsmooth(NJTSSWG * (1.0 + TNJTSSWG * (TRatio - 1.0)) - 0.01, 1.0e-3)+0.01;

PBS_t      = hypsmooth(PBS - TPB * delTemp - 0.01, 1.0e-3) + 0.01;

DELTA_t = 1.0 / ( hypsmooth((1.0 / DELTA_i) * (1.0 + TDELTA * delTemp) - 2.0 , 1.0e-3) + 2.0);

T2        = hypsmooth(IJTHSREV / Isbs - 10.0, 1.0e-3) + 10.0;
```

- However, some calls to hypsmooth () are still problematic:

```
T0      = hypsmooth((2.0 * phib + Vs * inv_Vt), 1.0e-3);
nq      = 1.0 + gam / (2.0 * sqrt(T0));

T4      = hypsmooth((1.0 + PDIBLCB_i * Vbsx), 1.0e-3);
T5      = 1.0 / T4;
```

- In the case when |x| is large, we can use Taylor series:

$$sqrt(1 + \varepsilon) = 1 + \frac{\varepsilon}{2} - \frac{\varepsilon^2}{8} + \cdots$$

$$sqrt(x * x + 4.0 * c * c) = |x| * sqrt(1.0 + 4.0 * c * c/(x * x))$$

$$sqrt(x * x + 4.0 * c * c) \approx |x| * (1 + 2 * (c/x)\text{^}2 - 2 * (c/x)\text{^}4)$$

$$sqrt(x * x + 4.0 * c * c) \approx |x| + |x| * 2 * (c/x)\text{^}2\ (1 - (c/x)\text{^}2)$$

- When x < 0, the leading 1 cancels the x in hypsmooth (x + |x| = 0)

$$\frac{\{x + sqrt(x * x + 4.0 * c * c)\}}{2} \approx |x| * (c/x)\text{^}2\ (1 - (c/x)\text{^}2\ )$$

- Also, the second term is not significant; omitting it gives a tiny error of < 1e-15 when x < -1e4 * c {or (c/x) <= 1e-8}. Therefore,

$$0.5 * \{x + sqrt(x * x + 4.0 * c * c)\} \approx |x| * (c/x)^2 = -\frac{c * c}{x}$$

BSIM-BULK 107.2.0 beta1

```
620  // Hyperbolic smoothing function
621  analog function real hypsmooth;
622      input x, c;
623      real x, c;
624      begin
625          hypsmooth = 0.5 * (x + sqrt(x * x + 4.0 * c * c));
626      end
627  endfunction
```

BSIM-BULK 107.2.0 beta2

```
620  // Hyperbolic smoothing function
621  analog function real hypsmooth;
622      input x, c;
623      real x, c;
624      begin
625          if (x < -1e4 *c) begin
626              hypsmooth = -c * c / x;
627          end else begin
628              hypsmooth = 0.5 * (x + sqrt(x * x + 4.0 * c * c));
629          end
630      end
631  endfunction
```

16. 2024bug7 (IITK/UCB): Addressing potential convergence warnings by using ln_one_plus_exp () function.

- Running Spectre simulation in diagnostic mode resulted in following potential warnings from lines 3355 and 3364:

```
230         WARNING (AHDLLINT-8009): "bsimbulk_beta1.va" 3355: NMOS:
231             Math function exp() value change between iterations is too large (2980.96), which might lead to convergence
232             difficulties.
233         WARNING (AHDLLINT-8009): "bsimbulk_beta1.va" 3364: NMOS:
234             Math function exp() value change between iterations is too large (2980.96), which might lead to convergence
235             difficulties.
```

BSIM-BULK 107.2.0 beta1

```
3351        T0   = AVDSX * Vdscv;
3352        if (T0 > `EXPL_THRESHOLD) begin
3353            T1 = T0;
3354        end else begin
3355            T1 = ln(1.0 + exp(T0));
3356        end
3357        Vdsx   = ((2.0 / AVDSX) * T1) - Vdscv - ((2.0 / AVDSX) * ln(2.0));
3358        Vbsxcv = -(Vscv + 0.5 * (Vdscv - Vdsx));
3359
3360        T0   = AVDSX * Vds;
3361        if (T0 > `EXPL_THRESHOLD) begin
3362            T1 = T0;
3363        end else begin
3364            T1 = ln(1.0 + exp(T0));
3365        end
3366        Vdsx = ((2.0 / AVDSX) * T1) - Vds - ((2.0 / AVDSX) * ln(2.0));
3367        Vbsx = -(Vs + 0.5 * (Vds - Vdsx));
```

- Using the `ln_one_plus_exp()` function instead of the above highlighted sections mitigates the error:

BSIM-BULK 107.2.0 beta2

```
3394
3395        T0   = AVDSX * Vdscv;
3396        T1 = ln_one_plus_exp(T0);
3397        Vdsx   = ((2.0 / AVDSX) * T1) - Vdscv - ((2.0 / AVDSX) * ln(2.0));
3398        Vbsxcv = -(Vscv + 0.5 * (Vdscv - Vdsx));
3399
3400        T0   = AVDSX * Vds;
3401        T1 = ln_one_plus_exp(T0);
3402        Vdsx = ((2.0 / AVDSX) * T1) - Vds - ((2.0 / AVDSX) * ln(2.0));
3403        Vbsx = -(Vs + 0.5 * (Vds - Vdsx));
```

17. 2024bug8 (UCB/IITK): Update to smooth macro definition

- Functionality of Smooth macro in BSIM-BULK 107.2.0 beta1 is same as that of hypsmooth function when the second argument of Smooth macro x0=0.

- Therefore, 2024 bug6 is relevant to Smooth macro as well.

- Smooth macro in BSIM-BULK 107.2.0 beta2 is updated as shown below.

```
// Smoothing function for (max of x, x0 with deltax)
`define Smooth(x, x0, deltax, xsmooth) \
    if ((x0 == 0.0) && ((x) < (-2500.0 * deltax))) begin \
        xsmooth = -deltax * deltax / (16.0 * (x)); \
    end else begin \
        xsmooth = 0.5 * (x + x0 + sqrt((x - x0) * (x - x0) + 0.25 * deltax * deltax)); \
    end \
```

18. 2024bug9 (UCB/IITK): Ensuring BETA1_i to be always non-negative number.

- In BSIM-BULK 107.2.0 beta1 code, Vdseffii expression is defined as shown below.

```
Vdssatii = (1.0 + BETA1_i * Vds) * Vdssat;
T7 = pow((Vds / Vdssatii) + 1.0e-6, 1.0 / DELTA_t);
T8 = pow(1.0 + T7, -DELTA_t);
Vdseffii = Vds * T8 ;
```

- In the above code segment, Vdssatii can become negative if BETA1_i is negative.
- In BSIM-BULK 107.2.0 beta1 code, BETA1 has been defined using MPRnb macro.
- In BSIM-BULK 107.2.0 beta2 code, BETA1 has been defined using MPRcz macro to ensure that BETA1 will not be negative.
- Moreover, in BSIM-BULK 107.2.0 beta2 code, as shown below, BETA1_i has been limited to ensure that its value will not be negative.

```
if (BETA1_i < 0.0) begin
    $strobe("Warning: BETA1_i = %e is negative, setting it to 0.", BETA1_i);
    BETA1_i = 0.0;
end
```

19. 2024bug10 (UCB/IITK): Drain-side and source-side drift resistance symmetry

- In BSIM-BULK 107.2.0 beta1 code, the modification to the drain-side drift resistance saturation current shown below is not applied to the source-side drift resistance saturation current.

```
T2 = 1.0 + Vsb_noswap / vbi_drift;
`Smooth(T2, 0.0, 0.05, T2)
T6 = (1.0 - DRB1 * ( sqrt(T2) - 1.0) - DRB2 * Vsb_noswap);
`Smooth(T6, 0.0, 0.05, T6)
idrift_sat_d = T6 * idrift_sat_d;
```

- To ensure symmetry, in BSIM-BULK 107.2.0 beta2 code, similar modification is also introduced in the source-side drift resistance saturation current as shown below.

```
T2 = 1.0 + Vsb_noswap / vbi_drift;
`Smooth(T2, 0.0, 0.05, T2)
T6 = (1.0 - DRB1 * ( sqrt(T2) - 1.0) - DRB2 * Vsb_noswap);
`Smooth(T6, 0.0, 0.05, T6)
idrift_sat_s = T6 * idrift_sat_s;
```

- Similarly, to ensure symmetry and continuity in CV calculations for HVCAPS=1, in BSIM-BULK 107.2.0 beta2 code, the expression for q_k is modified as shown below.

- BSIM-BULK 107.2.0 beta1

```
`BSIM_q(psip_k, phibHV, Vdcv / Vt, gamhv, 1.0, q_k)
```

- BSIM-BULK 107.2.0 beta2

```
`BSIM_q(psip_k, phibHV, Vdcv_noswap / Vt, gamhv, 1.0, q_k)
```

20. 2024bug11 (UCB/IITK): Corrected PARAM_Check file in QA package for PMOS transistor

- 044_DC_PARAM_check test in PMOS folder is mistakenly done for NMOS transistor.
- Value of **TYPE** parameter in PARAM_Check file of PMOS transistor in BSIM-BULK 107.2.0 beta1 QA package is **1**.
- In BSIM-BULK 107.2.0 beta2 QA package, value of **TYPE** parameter in PARAM_Check file of PMOS transistor has been correctly updated as **-1**.

- Therefore, the test 044_DC_PARAM_Check of PMOS transistor in BSIM-BULK 107.2.0 beta QA package will show backward incompatibility even though **the code is backward compatible**.

21. 2024bug12 (ADI): Instance parameters removed from model parameters list in QA package.

- In BSIM-BULK 107.2.0 beta1 QA package, instance parameters RGEOMOD and GEOMOD are mentioned as model parameters.
- In BSIM-BULK 107.2.0 beta2 QA package, these instance parameters RGEMOD and GEOMOD are removed from model parameters list.

22. 2024bug13 (ADI): Updated incorrect parameter description and parameter units.

- In BSIM-BULK 107.2.0 beta1 code, the description for parameter PTWGTL has a typo with word “scaling” incorrectly written as “acaling”. This has been corrected in BSIM-BULK 107.2.0 beta2 code.
- BSIM-BULK 107.2.0 beta1

```
`MPRnb( PTWGTL   ,0.0   ,"m"   ,"Length acaling parameter for PTWGT" )
```

- BSIM-BULK 107.2.0 beta2

```
`MPRnb( PTWGTL   ,0.0   ,"m"   ,"Length scaling parameter for PTWGT" )
```

- The description of parameter EUWLEXP in BSIM-BULK 107.2.0 beta1 is corrected in BSIM-BULK 107.2.0 beta2 as shown below.
- BSIM-BULK 107.2.0 beta1

```
`MPRoz( EUWLEXP   ,1.0   ,""   ,"Width-length dependence coefficient of EU" )
```

- BSIM-BULK 107.2.0 beta2

```
`MPRoz( EUWLEXP  ,1.0  ,""  ,"Width-length dependence exponent coefficient of EU" )
```

- The definitions for parameters WEB, WEC and SCREF are shown below.
- BSIM-BULK 107.2.0 beta1

```
`MPRnb( WEB    ,0.0     ,""              ,"Coefficient for SCB (> 0)" )
`MPRnb( WEC    ,0.0     ,""              ,"Coefficient for SCC (> 0)" )
`MPRoo( SCREF  ,1.0e-6  ,"m"  ,0.0  ,inf  ,"Reference distance to calculate SCA,SCB and SCC (< 0)" )
```

- In BSIM-BULK 107.2.0 beta2, the parameter definitions for WEB, WEC and SCREF have been updated.
- Parameter definitions of WEB and WEC have been changed to MPRcz and the parameter definition of SCREF has been changed to MPRoz, as shown below.
- BSIM-BULK 107.2.0 beta2

```
`MPRcz( WEB    ,0.0     ,""   ,"Coefficient for SCB" )
`MPRcz( WEC    ,0.0     ,""   ,"Coefficient for SCC" )
`MPRoz( SCREF  ,1.0e-6  ,"m"  ,"Reference distance to calculate SCA,SCB and SCC" )
```

- Units of the parameters VSATCVL, VSATCVW and VSATCVWL were incorrect in BSIM-BULK 107.2.0 beta1. These have been corrected in BSIM-BULK 107.2.0 beta2.
- BSIM-BULK 107.2.0 beta1

```
`MPRnb( VSATCVL   ,VSATL   ,"m^VSATLEXP"       ,"Length dependence coefficient of VSATCV" )
`MPRnb( VSATCVW   ,VSATW   ,"m^VSATWEXP"       ,"Width dependence coefficient of VSATCV" )
`MPRnb( VSATCVWL  ,VSATWL  ,"m^(2*VSATWLEXP)"  ,"Width-length dependence coefficient of VSATCV" )
```

- BSIM-BULK 107.2.0 beta2

```
`MPRnb( VSATCVL   ,VSATL   ,"m^VSATCVLEXP"       ,"Length dependence coefficient of VSATCV" )
`MPRnb( VSATCVW   ,VSATW   ,"m^VSATCVWEXP"       ,"Width dependence coefficient of VSATCV" )
`MPRnb( VSATCVWL  ,VSATWL  ,"m^(2*VSATCVWLEXP)"  ,"Width-length dependence coefficient of VSATCV" )
```

- A code snippet of BSIM-BULK 107.2.0 beta1 is shown below.
- BSIM-BULK 107.2.0 beta1

```
`MPRnb( C0SI1  ,0.0  ,"1/K"  ,"Temperature dependence of C0SI1" )
```

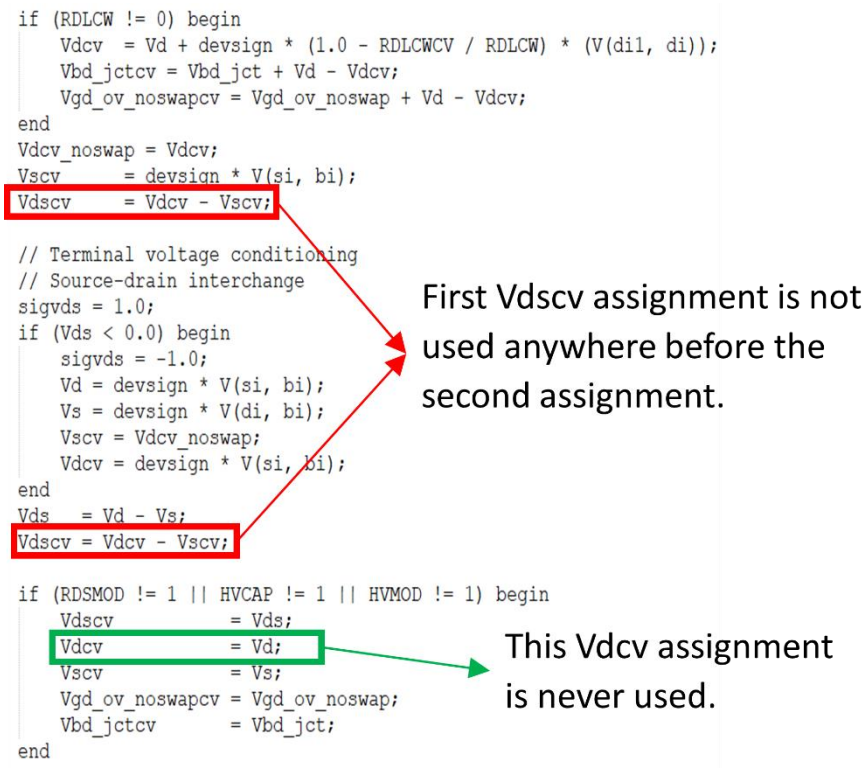
- In the above code snippet, C0SI1 is described as the temperature dependence parameter of C0SI1 instead of C0SI. This typo is corrected in BSIM-BULK 107.2.0 beta2 as shown below.
- BSIM-BULK 107.2.0 beta2

```
`MPRnb( C0SI1   ,0.0   ,"1/K"   ,"Temperature dependence of C0SI" )
```

- Parameter description of the parameters C0SISAT1, U0L, PSATL and PSATLEXP also had similar typos, and these were also corrected in BSIM-BULK 107.2.0 beta2.

23. 2024bug14 (UCB/IITK): Removed superfluous assignments.

- In BSIM-BULK 107.2.0 beta1 code, there were two superfluous assignments as shown below.



- In BSIM-BULK 107.2.0 beta2 code, we have modified the code segment to avoid superfluous assignments.

```
if (RDLCW != 0 && RDSMOD == 1 && HVCAP == 1 && HVMOD == 1) begin
    Vdcv  = Vd + devsign * (1.0 - RDLCWCV / RDLCW) * (V(di1, di));
    Vbd_jctcv = Vbd_jct + Vd - Vdcv;
    Vgd_ov_noswapcv = Vgd_ov_noswap + Vd - Vdcv;
end
Vdcv_noswap = Vdcv;
Vscv      = devsign * V(si, bi);

// Terminal voltage conditioning
// Source-drain interchange
sigvds = 1.0;
if (Vds < 0.0) begin
    sigvds = -1.0;
    Vd = devsign * V(si, bi);
    Vs = devsign * V(di, bi);
    Vscv = Vdcv_noswap;
    Vdcv = devsign * V(si, bi);
end
Vds   = Vd - Vs;
Vdscv = Vdcv - Vscv;
```

24. 2024bug15 (ADI): Ensured the manual mentions the correct model name.

- In the BSIM-BULK 107.2.0 beta1 manual, the model name had been incorrectly referenced as BSIM6 in few places.
- In the new BSIM-BULK 107.2.0 beta2 manual, the model name has been correctly referenced as BSIM-BULK in all the places.

BSIM-BULK 107.2.0_beta3

H. Description of enhancements:

1. 2024enh5 (ADI): Add more QA tests to cover all parameters and MODs

- Used check_parameter_coverage.py file to add missing tests in the new QA package.
- The number of tests has increased from 133 to 242.

I. Description of bug-fixes:

2. 2024bug16 (ADI): Missing reference data in the QA package for some QA tests

- Test 091_AC_Vd_b_HVMOD1_WL for the PMOS transistor produced only 23 lines of reference data. Expected 67 lines of reference data.
- Similarly tests 094_AC_Vd_b_HVCAP1_WL and 097_AC_Vd_b_HVCAPS1_WL for the PMOS transistor produced only 34 and 23 lines of reference data respectively. Expected 67 lines of reference data.
- QA test results were generated with the latest version of the reference simulator to solve the problem.
- In the latest QA package, reference data for all the tests are present.

3. 2024bug17 (Infineon): Ensure CMD1 and CMS1 parameters have impact only when IIMOD=1.

- CMD1 and CMS1 parameters are introduced to model conductivity modulation effect which happens only when there is impact ionization in the drift region.
- Therefore, the parameters CMD1 and CMS1 should impact drift region saturation current only when IIMOD=1.
- Modified the code as shown below

BSIM-BULK 107.2.0 beta2

```
idrift_sat_d = T11 * NDRIFTD * T9 * (1 + devsign * CMD1 * pow(Vb_cm, CMD2));
```

BSIM-BULK 107.2.0 beta3

```
if (IIMOD != 0) begin
    idrift_sat_d = T11 * NDRIFTD * T9 * (1 + devsign * CMD1 * pow(Vb_cm, CMD2));
end else begin
    idrift_sat_d = T11 * NDRIFTD * T9;
end
```

BSIM-BULK 107.2.0 beta2

```
idrift_sat_s = T11 * NDRIFTS * (1 + devsign * CMS1 * pow(Vb_cm, CMS2));
```

BSIM-BULK 107.2.0 beta3

```
if (IIMOD != 0) begin
    idrift_sat_s = T11 * NDRIFTS * (1 + devsign * CMS1 * pow(Vb_cm, CMS2));
end else begin
    idrift_sat_s = T11 * NDRIFTS;
end
```