Summary of Changes from BSIM-BULK107.2.0 Beta 1 to BSIM-BULK107.2.0 Beta 2:

BSIM Group, IIT Kanpur, UC Berkeley

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

A. Summary of enhancements:

1. **2023enh3 (ADI):** Fitting flexibility of Cgg in moderate inversion.
2. **2023enh4 (Infineon):** Non-linear Vd dependency for Intrinsic Impact Ionization model.
3. **2023enh5 (Infineon):** Non-linear Vd dependency for drift-region Impact Ionization model.
4. **2023enh6 (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.

B. Summary of bug-fixes:

9. **2023bug13 (Infineon):** Smoothness requirement in Id-Vg for very high Vg and high Vd
10. **2024bug1 (ADI):** Removing the 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/incorrect OP descriptions.
15. **2024bug6 (ADI):** Update to hypsmooth () function.
16. **2024bug7 (IITK/UCB):** 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.

C. Description of enhancements:

1. 2023enh3 (ADI): Fitting flexibility of Cgg in moderate inversion.

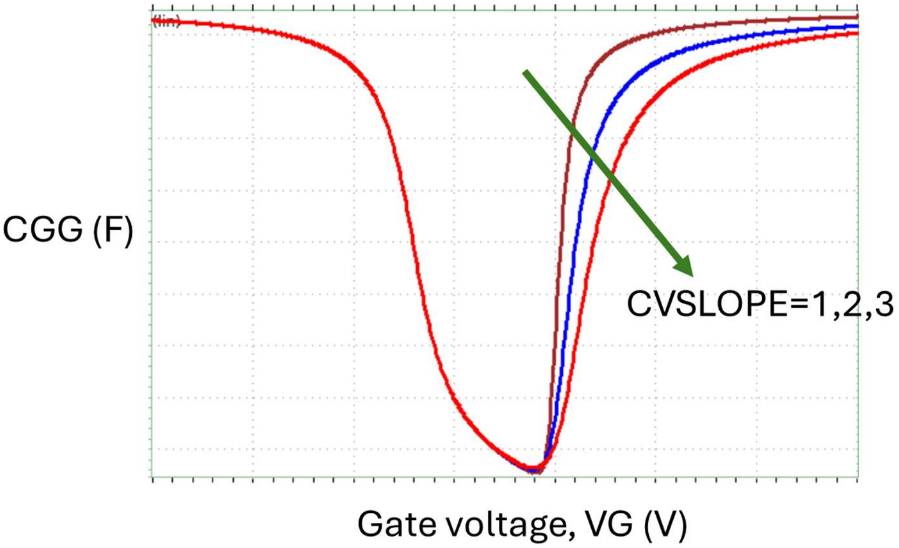
- The equation solved for computing the normalized inversion charge, q_i, in BSIM-BULK 107.2.0 beta1 is shown below.

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

- 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. 2023enh4 (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:

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

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

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

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

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

- 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. 2023enh5 (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:

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

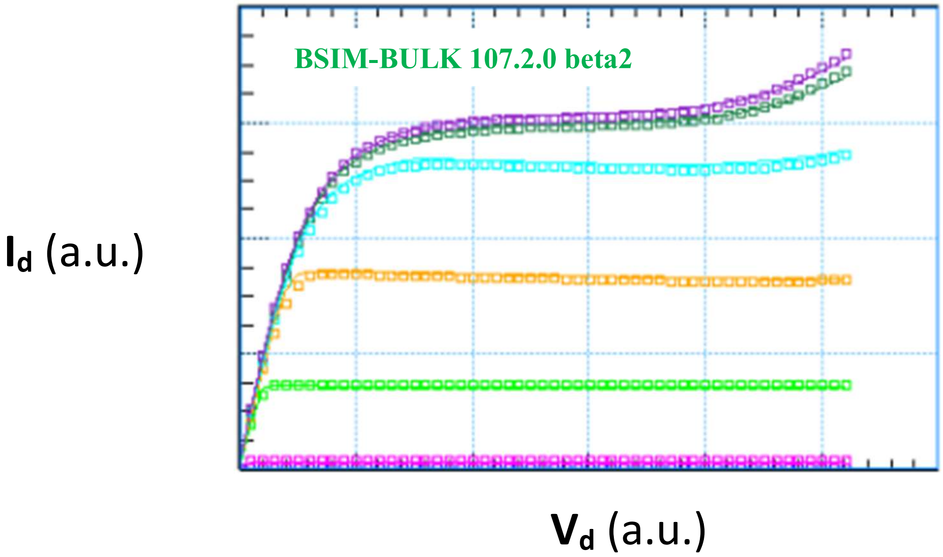
$$T5 = \boldsymbol{ALPHADR3} * VDDROP + \boldsymbol{ALPHADR4} * VDDROP^{\boldsymbol{DREXP}}$$

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

- 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. 2023enh6 (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.

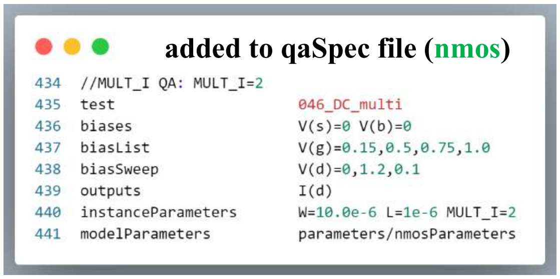
- These were set to default values.

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

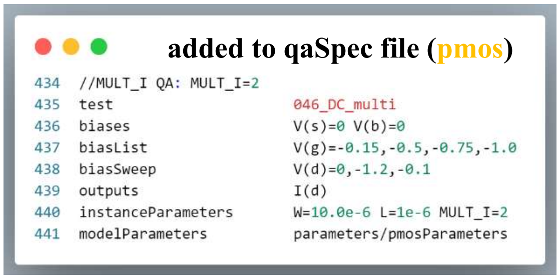
- In BSIM-BULK107.2.0 beta 1, there was no QA test using RDSMOD=1.
- In BSIM-BULK107.2.0 beta 2, QA tests 48 to 66 are added (in nmos and pmos qaSpec files).
- QA tests 48 to 66 in BSIM-BULK107.2.0 beta 2 are same as the QA tests 1 to 19 except that RDSMOD=1.

7. 2024enh3 (ADI): Added the QA tests for new MULT parameters.

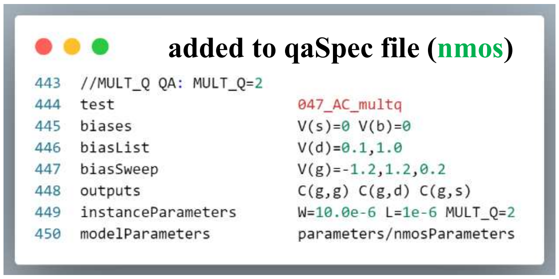
- In BSIM-BULK107.2.0 beta2, following three QA tests were added (in nmos and pmos qaSpec files) for the new MULT parameters (MULT_I, MULT_Q, MULT_FN):


added to qaSpec file (nmos)

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

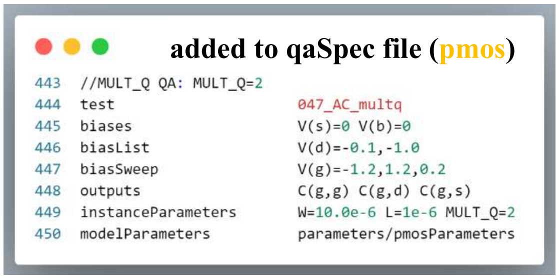

added to qaSpec file (pmos)

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



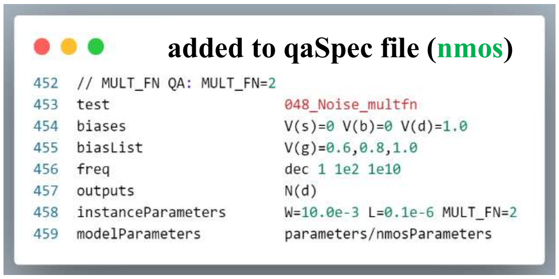
added to qaSpec file (nmos)

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



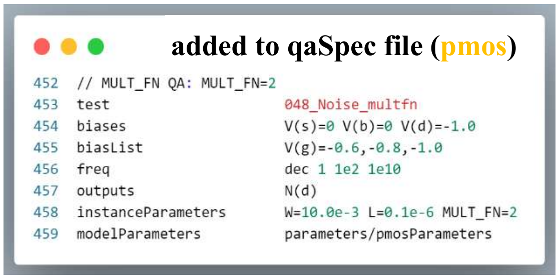
added to qaSpec file (pmos)

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



added to qaSpec file (nmos)

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



added to qaSpec file (pmos)

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

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

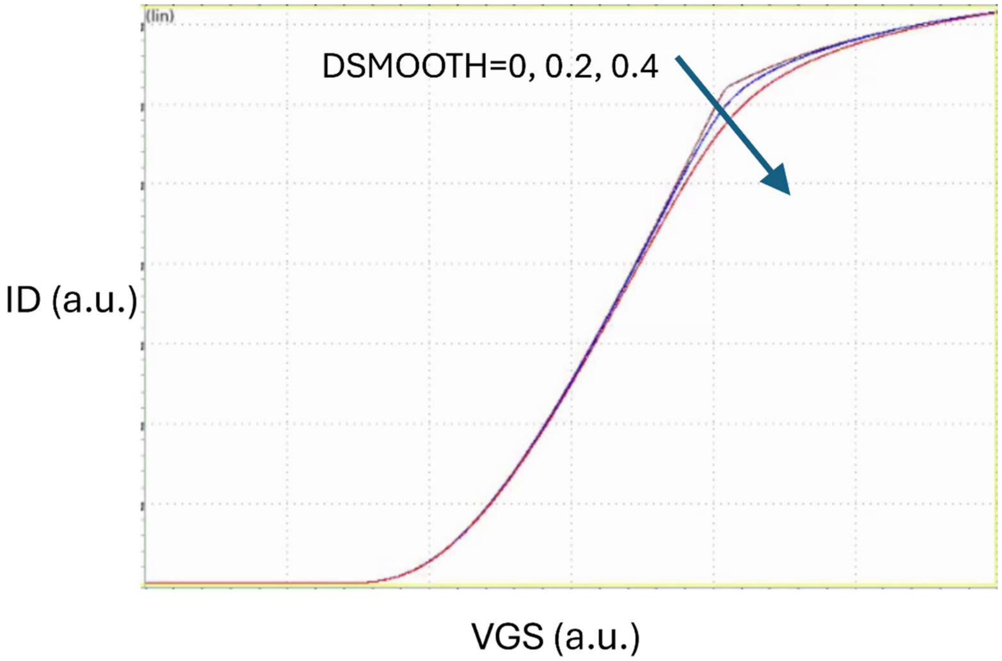
- Added QA tests for RDSMOD=0, HVMOD=1, HVCAP=1, HVCAPS=1, RGATEMOD=1, RGATEMOD=2, RBODYMOD=1, RBODYMOD=2, RBODYHVMOD=1, CVMOD=1, COVMOD=0, COVMOD=1 and SHMOD=1.
- Total number of QA tests have increased from 44 in BSIM-BULK 107.2.0 beta1 to 133 in BSIM-BULK 107.2.0 beta2.

D. Description of bug-fixes:

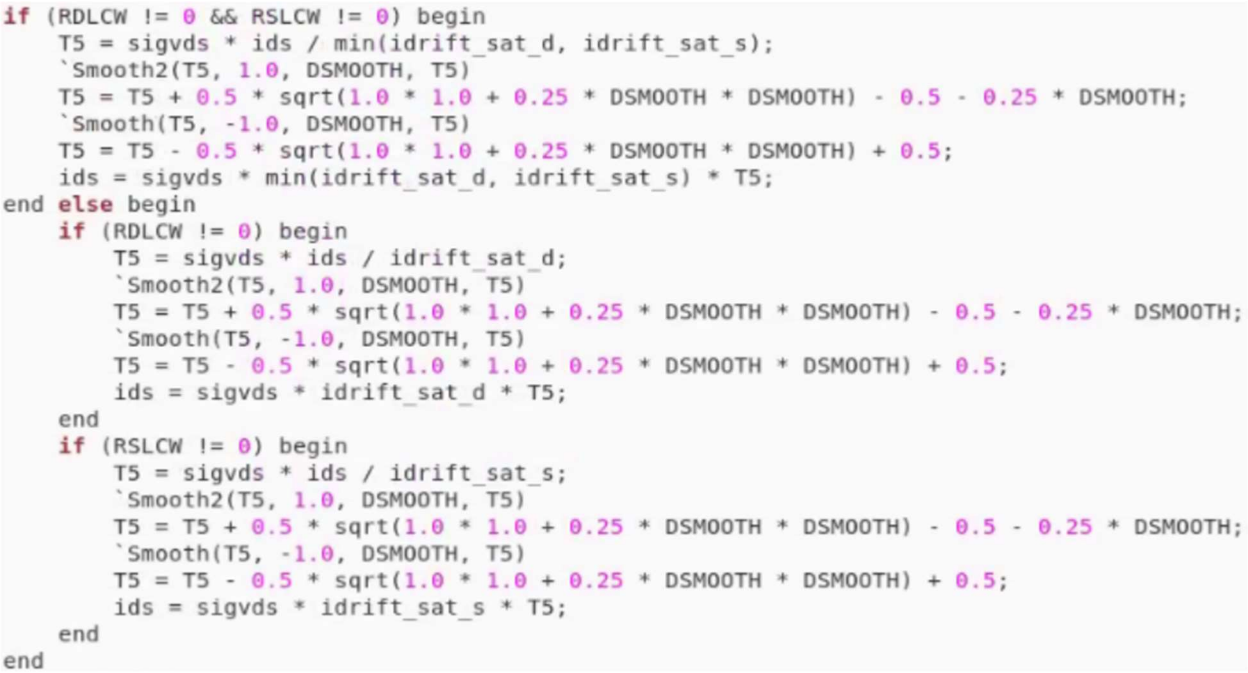
9. 2023bug13 (Infineon): Smoothness requirement in Id-Vg for very high Vg and high Vd

- 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.

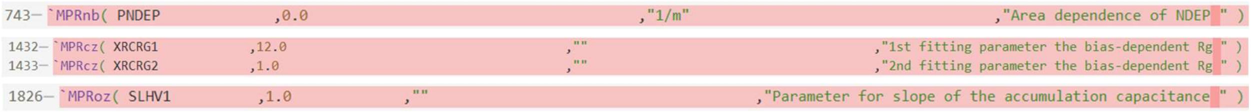


```
if (RDLCW != 0 && RSLCW != 0) begin
    T5 = sigvds * ids / min(idrift_sat_d, idrift_sat_s);
    `Smooth2(T5, 1.0, DSMOOTH, T5)
    T5 = T5 + 0.5 * sqrt(1.0 * 1.0 + 0.25 * DSMOOTH * DSMOOTH) - 0.5 - 0.25 * DSMOOTH;
    `Smooth(T5, -1.0, DSMOOTH, T5)
    T5 = T5 - 0.5 * sqrt(1.0 * 1.0 + 0.25 * DSMOOTH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

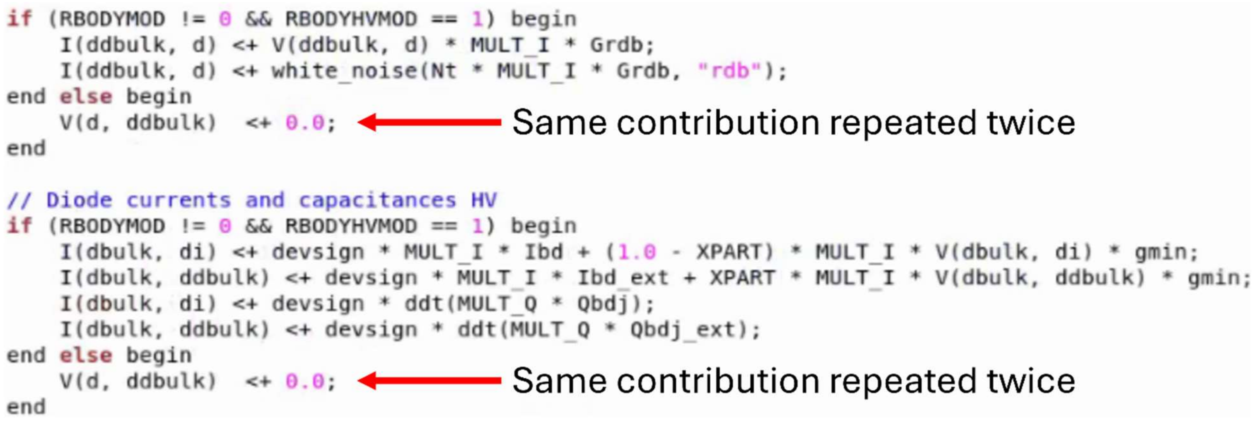
- Low value of MDRIFT can cause numerical round-off issue resulting in discontinuity in the drain current.

- 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



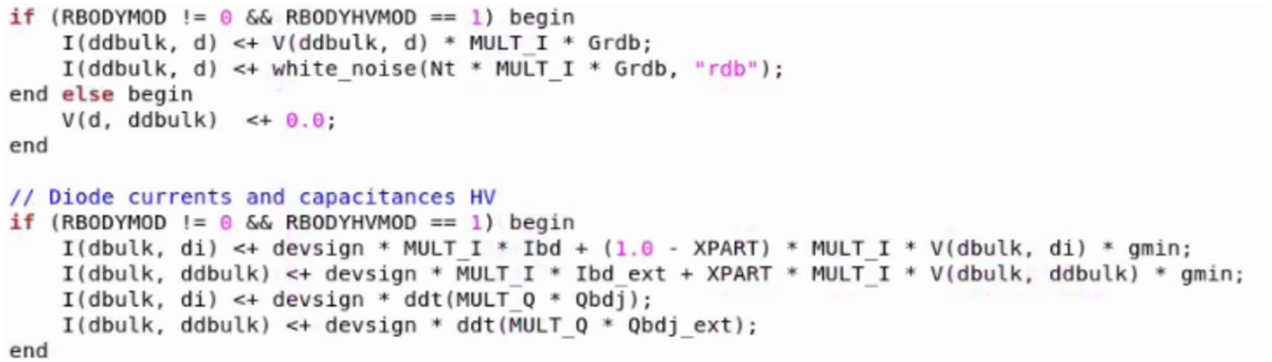
```
if (RBODYMOD != 0 && RBODYHVMOD == 1) begin
    I(ddbulk, d) <+ V(ddbulk, d) * MULT_I * Grdb;
    I(ddbulk, d) <+ white_noise(Nt * MULT_I * Grdb, "rdb");
end else begin
    V(d, ddbulk)  <+ 0.0;
end

// 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.

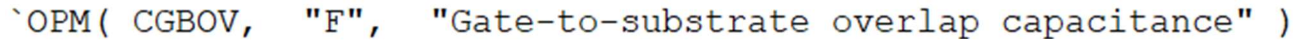
14. 2024bug5 (ADI): Updated the missing/incorrect OP descriptions.

- 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" )
```

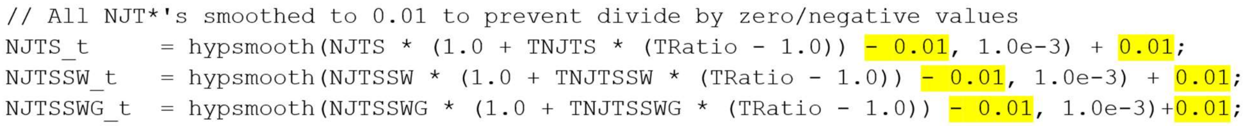
BSIM-BULK 107.2.0 beta2



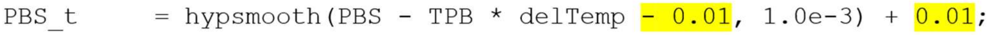
```
`OPM( CGBOV,  "F",  "Gate-to-substrate overlap capacitance" )
```

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

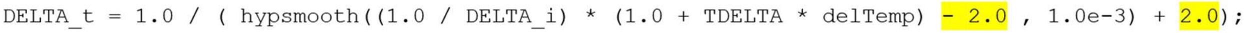
- BSIM-BULK 107.2.0 beta1 uses hypsmooth (x, c) to prevent division by zero.
- When |x| >> c, numerical problems (roundoff) can occur. Because in this case (x * x + 4.0 * c * c) is numerically equal to |x|.
- Therefore, hypsmooth (x, c) is numerically exactly 0 for large negative x.
- Although extra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



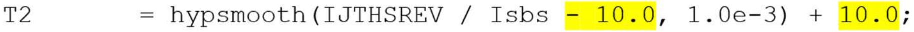
```
// All NJT*'s smoothed to 0.01 to prevent divide by zero/negative values
NJTS_t     = hypsmooth(NJTS * (1.0 + TNJTS * (TRatio - 1.0)) - 0.01, 1.0e-3) + 0.01;
NJTSSW_t   = hypsmooth(NJTSSW * (1.0 + TNJTSSW * (TRatio - 1.0)) - 0.01, 1.0e-3) + 0.01;
NJTSSWG_t  = hypsmooth(NJTSSWG * (1.0 + TNJTSSWG *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



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

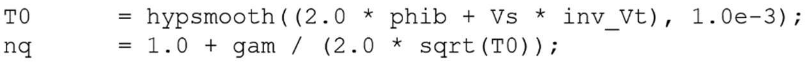


```
DELTA_t = 1.0 / ( hypsmooth((1.0 / DELTA_i) * (1.0 + TDELTA * delTemp) - 2.0 , 1.0e-3) + 2.0);
```

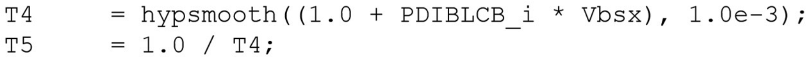


```
T2        = hypsmooth(IJTHSREV / Isbs - 10.0, 1.0e-3) + 10.0;
```

- 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:

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

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

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

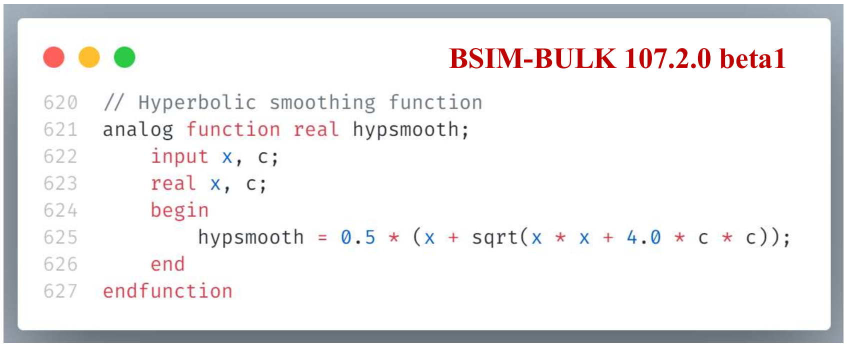
$$sqrt(x\ *\ x\ +\ 4.0\ *\ c\ *\ c) \approx\ |x| + |x| * 2 * (c/x)\text{^}2\ \ (1\ -\ (c/x)\text{^}2\ \ )$$

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

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

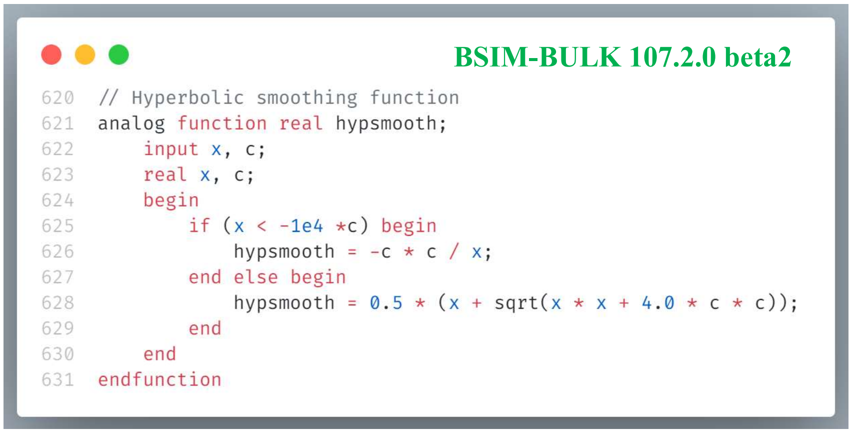
- Also, the second term is not significant; omitting it gives a tiny error of < 1e-15 when x < -1e4 * c {or (c/x) <= 1e-8}. Therefore,

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



BSIM-BULK 107.2.0 beta1

```
// Hyperbolic smoothing function
analog function real hypsmooth;
    input x, c;
    real x, c;
    begin
        hypsmooth = 0.5 * (x + sqrt(x * x + 4.0 * c * c));
    end
endfunction
```

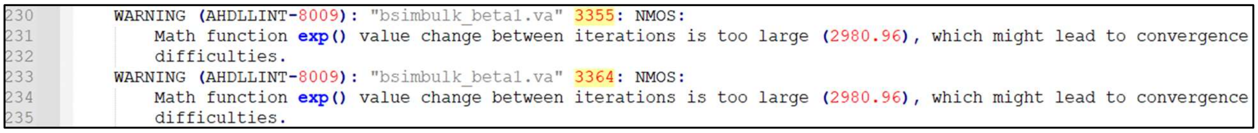


BSIM-BULK 107.2.0 beta2

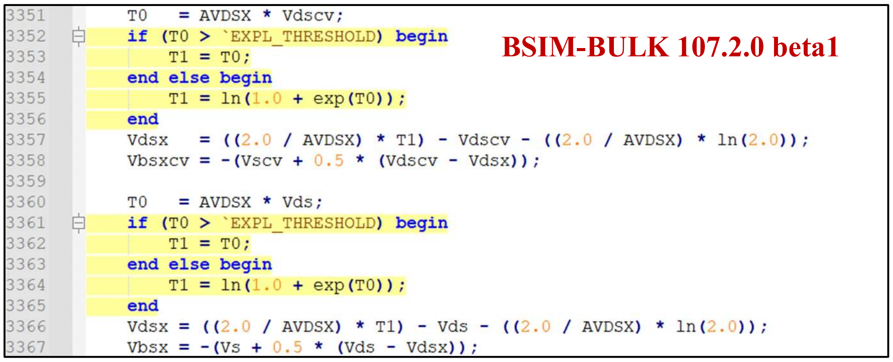
```
// Hyperbolic smoothing function
analog function real hypsmooth;
    input x, c;
    real x, c;
    begin
        if (x < -1e4 *c) begin
            hypsmooth = -c * c / x;
        end else begin
            hypsmooth = 0.5 * (x + sqrt(x * x + 4.0 * c * c));
        end
    end
endfunction
```

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:



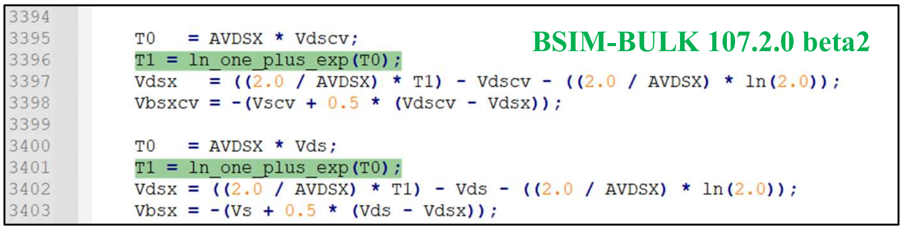
```
WARNING (AHDLLINT-8009): "bsimbulk_beta1.va" 3355: NMOS:
    Math function exp() value change between iterations is too large (2980.96), which might lead to convergence
    difficulties.
WARNING (AHDLLINT-8009): "bsimbulk_beta1.va" 3364: NMOS:
    Math function exp() value change between iterations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

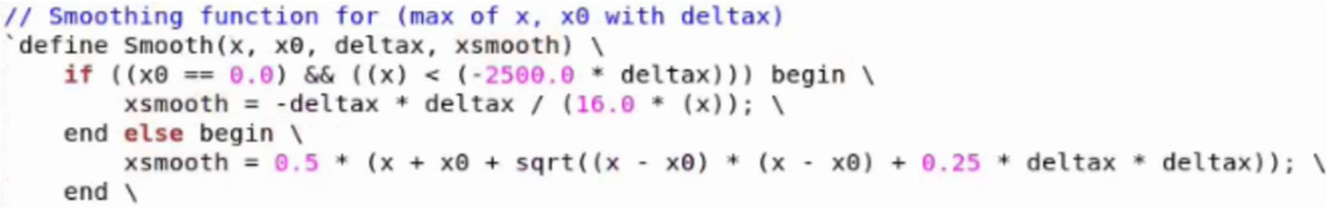
```

T0   = AVDSX * Vdscv;
T1 = ln_one_plus_exp(T0);
Vdsx    = ((2.0 / AVDSX) * T1) - Vdscv - ((2.0 / AVDSX) * ln(2.0));
Vbsxcv = -(Vscv + 0.5 * (Vdscv - Vdsx));

T0   = AVDSX * Vds;
T1 = ln_one_plus_exp(T0);
Vdsx = ((2.0 / AVDSX) * T1) - Vds - ((2.0 / AVDSX) * ln(2.0));
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.

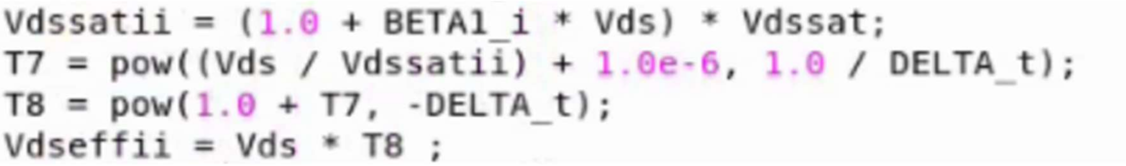
- Smooth macro in BSIM-BULK 107.2.0 beta2 is updated as shown below.



```
// 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 \
```

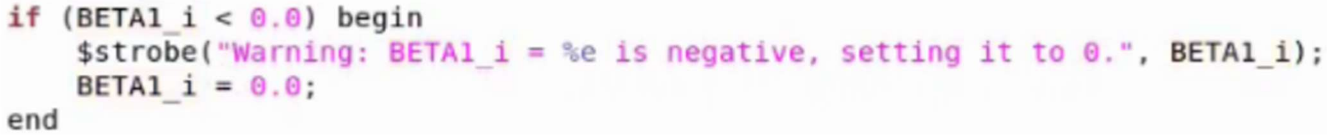
18. 2024bug9 (UCB/IITK): Ensuring BETA1_i to be always non-negative number.

- 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 ;
```

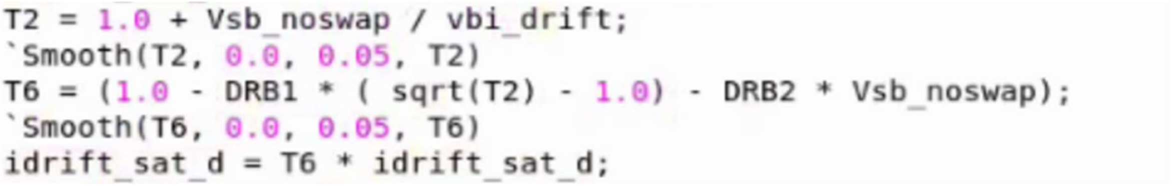
- In the above code segment, Vdssatii can become negative if BETA1_i is negative.
- In BSIM-BULK 107.2.0 beta1 code, BETA1 has been defined using MPRnb macro.
- In BSIM-BULK 107.2.0 beta2 code, BETA1 has been defined using MPRcz macro to ensure that BETA1 will not be negative.
- Moreover, in BSIM-BULK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

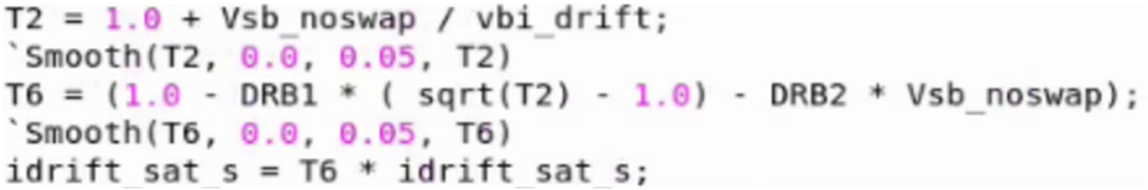
19. 2024bug10 (UCB/IITK): Drain-side and source-side drift resistance symmetry

- 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;
```

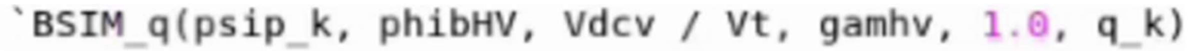
- To ensure symmetry, in BSIM-BULK 107.2.0 beta2 code, similar modification is also introduced in the source-side drift resistance saturation current as shown below.



```
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;
```

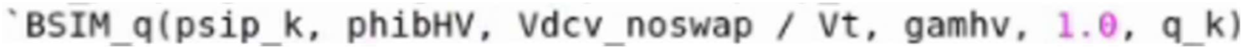
- Similarly, to ensure symmetry and continuity in CV calculations for HVCAPS=1, in BSIM-BULK 107.2.0 beta2 code, the expression for q_k is modified as shown below.

- 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**.

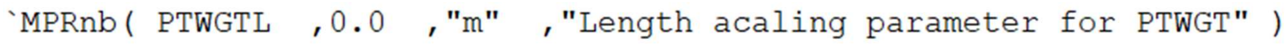
- Therefore, the test 044_DC_PARAM_Check of PMOS transistor in BSIM-BULK 107.2.0 beta QA package will show backward incompatibility even though **the code is backward compatible**.

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.

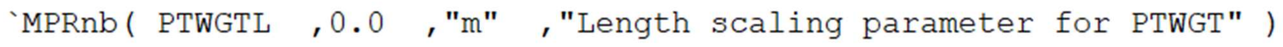
22. 2024bug13 (ADI): Updated incorrect parameter description and parameter units.

- 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



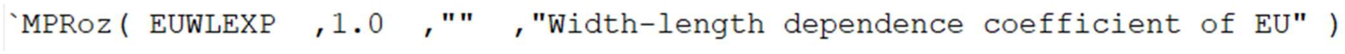
```
`MPRnb( PTWGTL  ,0.0  ,"m"  ,"Length acaling parameter for PTWGT" )
```

- BSIM-BULK 107.2.0 beta2



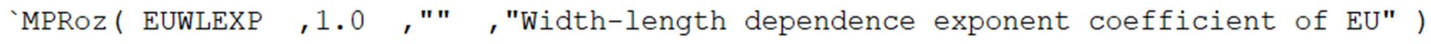
```
`MPRnb( PTWGTL  ,0.0  ,"m"  ,"Length scaling parameter for PTWGT" )
```

- 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



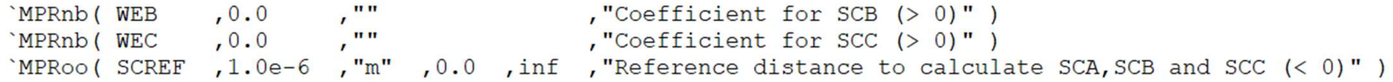
```
`MPRoz( EUWLEXP  ,1.0  ,""  ,"Width-length dependence coefficient of EU" )
```

- <u>BSIM-BULK 107.2.0 beta2</u>



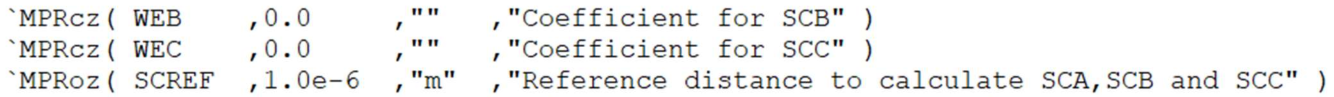
```
`MPRoz( EUWLEXP  ,1.0  ,""  ,"Width-length dependence exponent coefficient of EU" )
```

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



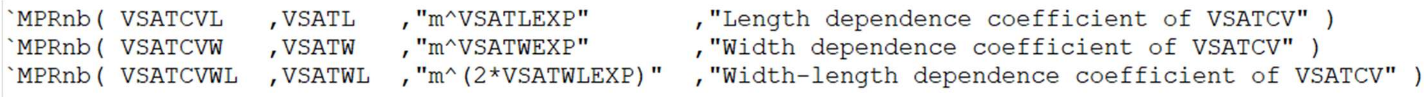
```
`MPRnb( WEB    ,0.0     ,""                  ,"Coefficient for SCB (> 0)" )
`MPRnb( WEC    ,0.0     ,""                  ,"Coefficient for SCC (> 0)" )
`MPRoo( SCREF  ,1.0e-6  ,"m"  ,0.0  ,inf  ,"Reference distance to calculate SCA,SCB and SCC (< 0)" )
```

- 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.
- <u>BSIM-BULK 107.2.0 beta2</u>



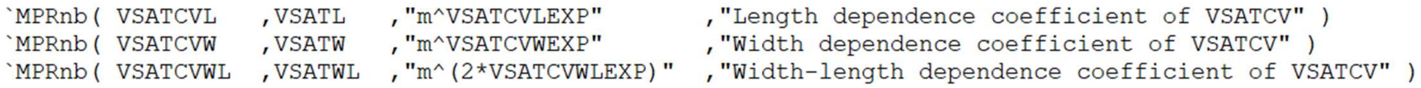
```
`MPRcz( WEB    ,0.0     ,""   ,"Coefficient for SCB" )
`MPRcz( WEC    ,0.0     ,""   ,"Coefficient for SCC" )
`MPRoz( SCREF  ,1.0e-6  ,"m"  ,"Reference distance to calculate SCA,SCB and SCC" )
```

- 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.
- <u>BSIM-BULK 107.2.0 beta1</u>



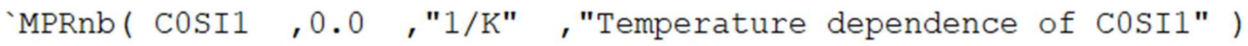
```
`MPRnb( VSATCVL   ,VSATL   ,"m^VSATLEXP"        ,"Length dependence coefficient of VSATCV" )
`MPRnb( VSATCVW   ,VSATW   ,"m^VSATWEXP"        ,"Width dependence coefficient of VSATCV" )
`MPRnb( VSATCVWL  ,VSATWL  ,"m^(2*VSATWLEXP)"   ,"Width-length dependence coefficient of VSATCV" )
```

- <u>BSIM-BULK 107.2.0 beta2</u>



```
`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.
- <u>BSIM-BULK 107.2.0 beta1</u>



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

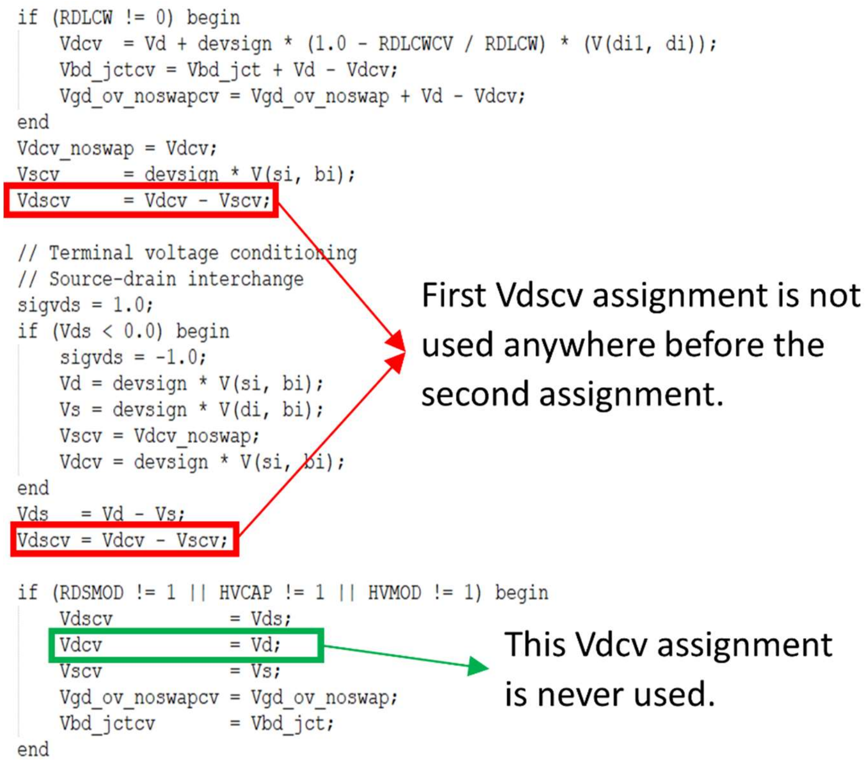
- In the above code snippet, C0SI1 is described as the temperature dependence parameter of C0SI1 instead of C0SI. This typo is corrected in BSIM-BULK 107.2.0 beta2 as shown below.
- BSIM-BULK 107.2.0 beta2

```
`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.

25. 2024bug14 (UCB/IITK): Removed superfluous assignments.

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

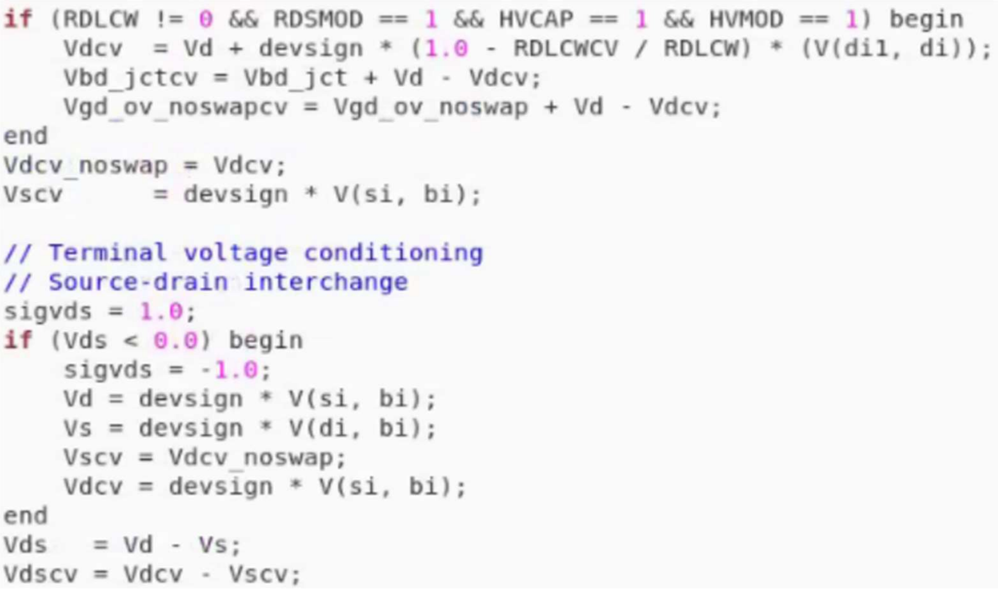


```
if (RDLCW != 0) begin
    Vdcv  = Vd + devsign * (1.0 - RDLCWCV / RDLCW) * (V(di1, di));
    Vbd_jctcv = Vbd_jct + Vd - Vdcv;
    Vgd_ov_noswapcv = Vgd_ov_noswap + Vd - Vdcv;
end
Vdcv_noswap = Vdcv;
Vscv       = devsign * V(si, bi);
Vdscv      = Vdcv - Vscv;

// 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;

if (RDSMOD != 1 || HVCAP != 1 || HVMOD != 1) begin
    Vdscv           = Vds;
    Vdcv            = Vd;
    Vscv            = Vs;
    Vgd_ov_noswapcv = Vgd_ov_noswap;
    Vbd_jctcv       = Vbd_jct;
end
```

First Vdscv assignment is not used anywhere before the second assignment.

This Vdcv assignment is never used.

- 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;
```

26. 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.