Summary of Changes from BSIM-BULK107.2.0 Beta0_1 to BSIM-BULK107.2.0 Beta0_2:

BSIM Group, IIT Kanpur, UC Berkeley

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

A. Summary of Enhancements:

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

B. Summary of bug-fixes:

3. **2023bug3 (ADI):** Issue in C_{GD} reciprocity while exercising QM parameters.
4. **2023bug4 (ADI):** MULT implementation correction.
5. **2023bug5 (ADI):** Correction in units of binning parameters.
6. **2023bug6 (ADI):** Removing variables that were superfluously assigned.
7. **2023bug7 (ADI):** Addressing the bias-dependent $strobe warnings reported by VAMPyRE.

C. Description of Enhancements:

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

- In BSIM-BULK107.2.0 Beta0_1, 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-BULK107.2.0 Beta0_1 is:

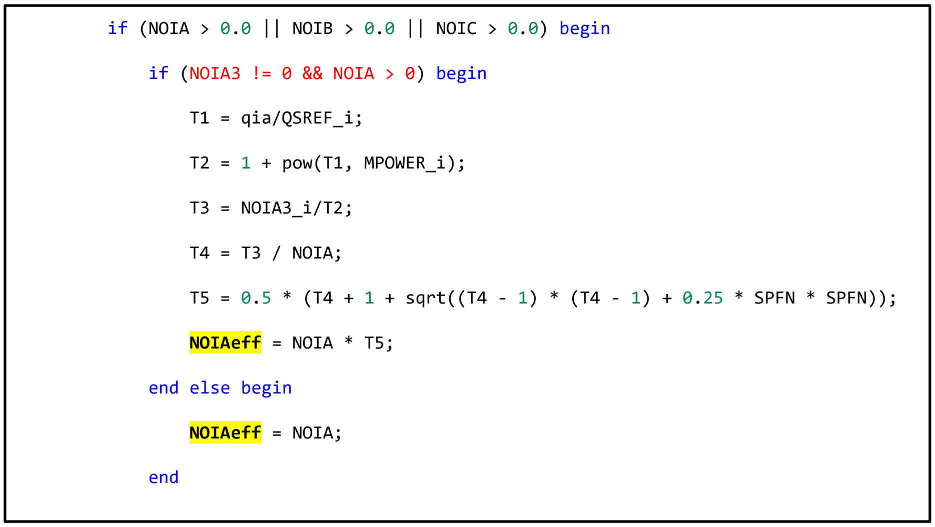
$$\boxed{S_{wi} = \frac{\boldsymbol{NOIA}. kT. I_{ds}^{2}}{W_{eff}. NF. L_{eff,noi}\,.\,10^{10}. N^{*2}}}$$

- In BSIM-BULK107.2.0 Beta0_2, a bias dependent effective NOIA is introduced to model the noise behavior in weak inversion:

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

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

- 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**.

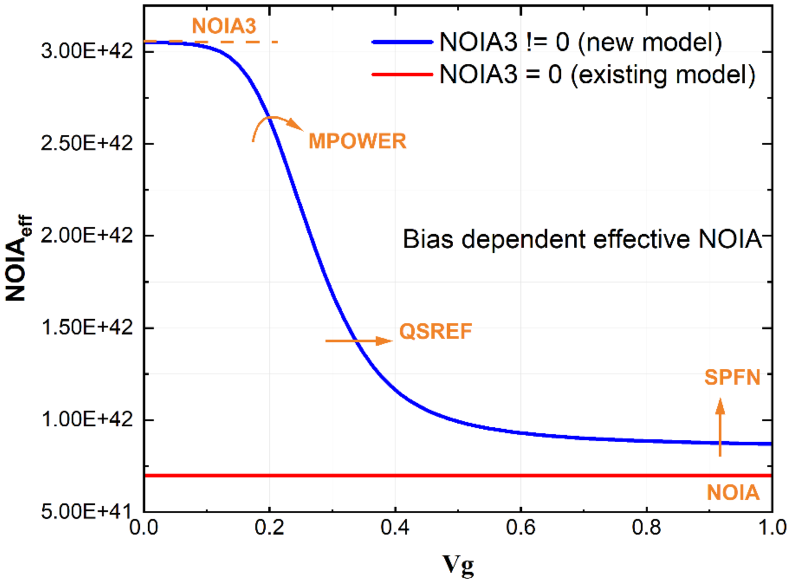


Fig. 1: A bias dependent NOIA is introduced when NOIA3 > 0 to model noise behavior in weak inversion region.

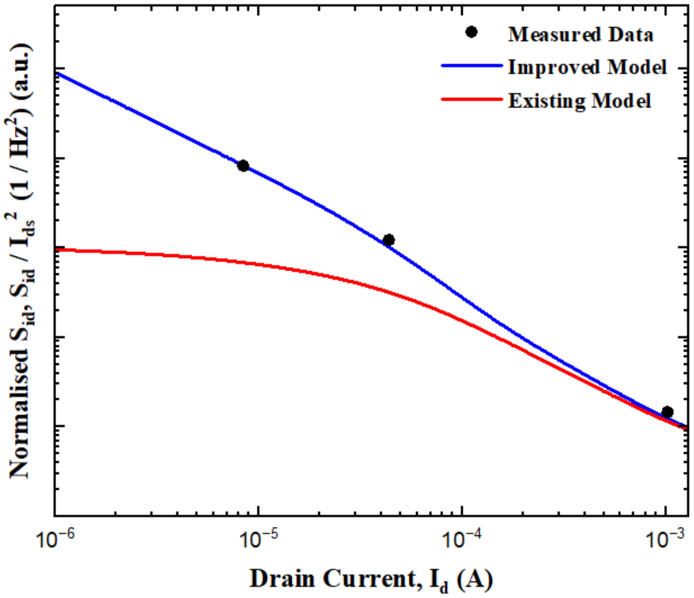


Fig. 2: New model is able to match the Sid/Id^2 trend vs current data in weak inversion region which existing model failed to capture.

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 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.
  - $V(di)_{CV} = V(di)_{IV} + devsign * (1.0 - RDLCWCV / RDLCW) * V(di1, di)$
- This new enhancement provides more flexibility in capacitance fitting without affecting the current characteristic fitting.
- An example case is shown below. Fig. 3 shows the capacitance plots using BSIM-BULK 107.2.0 beta0_1 and Fig.4 shows the same using the new BSIM-BULK 107.2.0 beta0_2 code.

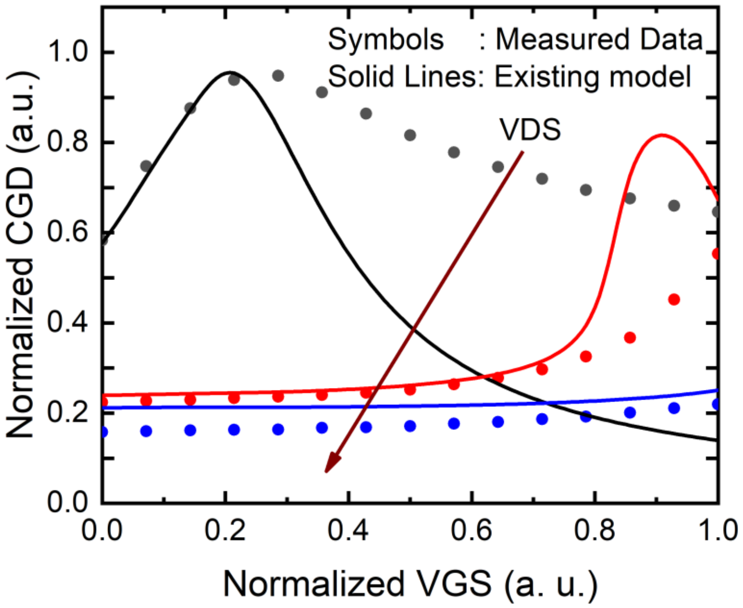


Fig. 3: Capacitance plots obtained using BSIM-BULK 107.2.0beta0_1 code.

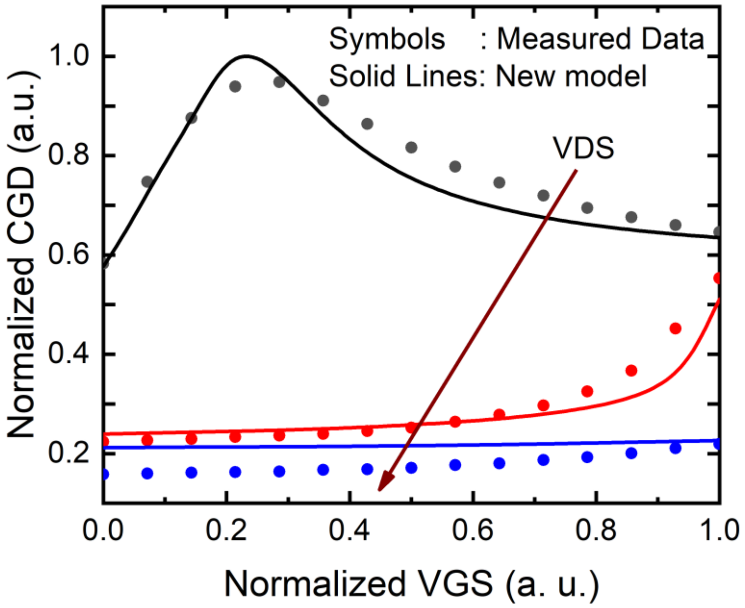


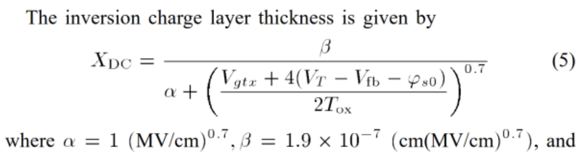
Fig. 4: Capacitance plots obtained using BSIM-BULK 107.2.0beta0_2 code.

D. Description of Bug Fixes:

3. 2023bug3 (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

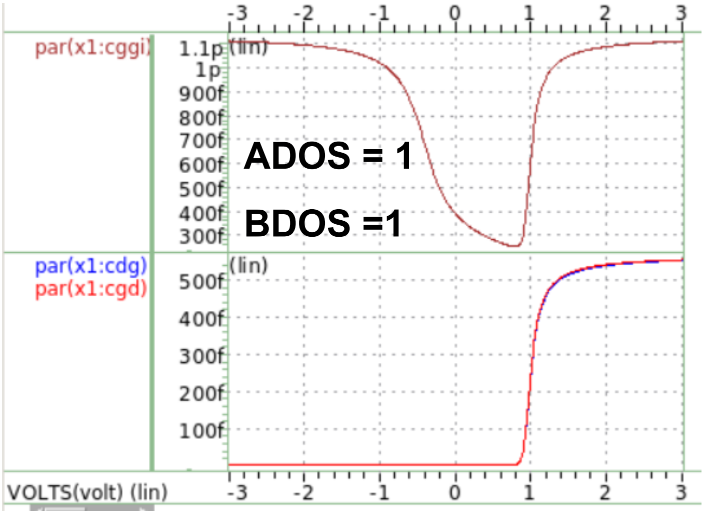
$$X_{\rm DC} = \frac{\beta}{\alpha + \left(\frac{V_{gtx} + 4(V_T - V_{\rm fb} - \varphi_{s0})}{2T_{\rm ox}}\right)^{0.7}} \qquad (5)$$

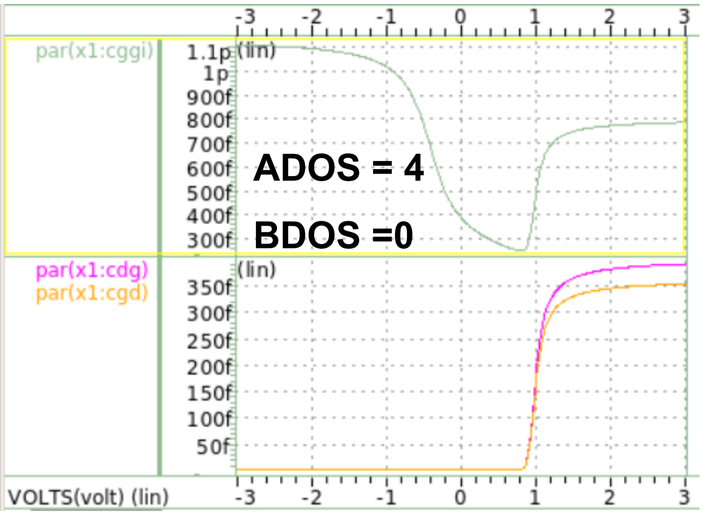
where $\alpha = 1\ ({\rm MV/cm})^{0.7}$, $\beta = 1.9 \times 10^{-7}\ ({\rm cm(MV/cm)}^{0.7})$, and

Parameterized expression in BSIM-BULK

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

- 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.

4. 2023bug4 (ADI): MULT implementation correction.

- The VAMPyRE v1.9.2 reported some errors (missing/extra contributions) in the BSIM-BULK107.2.0 Beta0_1.
- The reported errors were addressed in BSIM-BULK107.2.0 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

5. 2023bug5 (ADI): Correction in units of binning parameters.

- The VAMPyRE v1.9.2 reported some inconsistency in units of some binning parameters in the BSIM-BULK107.2.0 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-BULK107.2.0 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-BULK107.2.0 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.

6. 2023bug6 (ADI): Removing superfluous assignment of variables.

- The VAMPyRE v1.9.2 reported superfluous assignment of some variables in the BSIM-BULK107.2.0 Beta0_1.

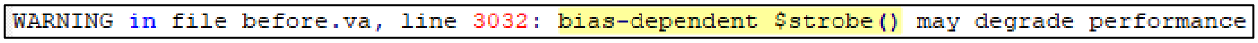
Following is a list of assignments/initialization that were dropped in BSIM-BULK107.2.0 Beta0_2:

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)**

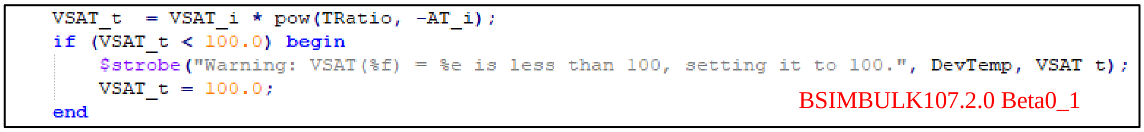
7. 2023bug7 (ADI): Addressing the bias dependent $strobe warnings reported by VAMPyRE.

- The VAMPyRE v1.9.2 reported some bias-dependent $strobe conditions in the BSIM-BULK107.2.0 Beta0_1.

○



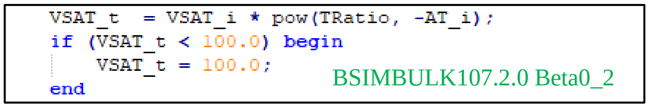
```
WARNING in file before.va, line 3032: bias-dependent $strobe() may degrade performance
```



```
    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

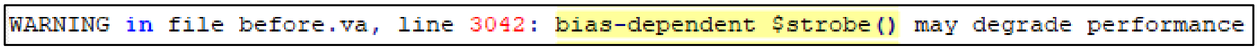
The $strobe statement was removed in this case as follows:



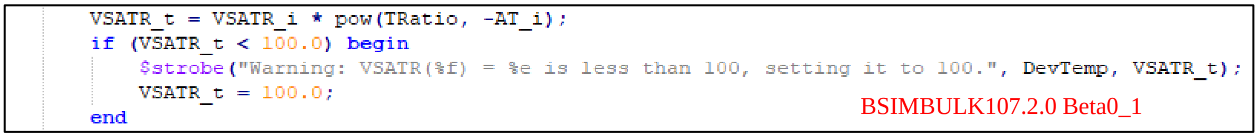
```
    VSAT_t  = VSAT_i * pow(TRatio, -AT_i);
    if (VSAT_t < 100.0) begin
        VSAT_t = 100.0;
    end
```

BSIMBULK107.2.0 Beta0_2

○



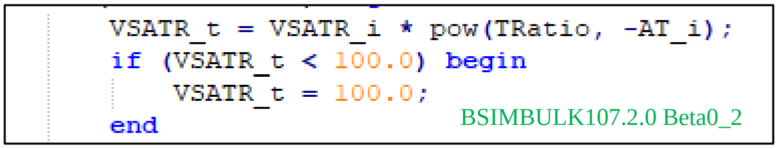
```
WARNING in file before.va, line 3042: bias-dependent $strobe() may degrade performance
```



```
        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

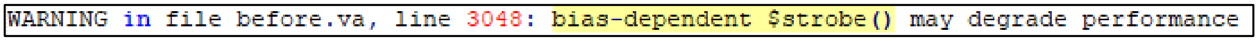
The $strobe statement was removed in this case as follows:



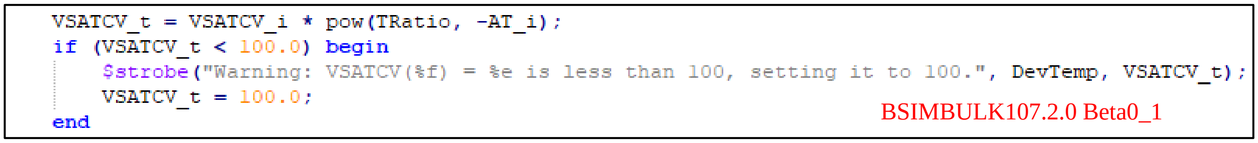
```
    VSATR_t = VSATR_i * pow(TRatio, -AT_i);
    if (VSATR_t < 100.0) begin
        VSATR_t = 100.0;
    end
```

BSIMBULK107.2.0 Beta0_2

○



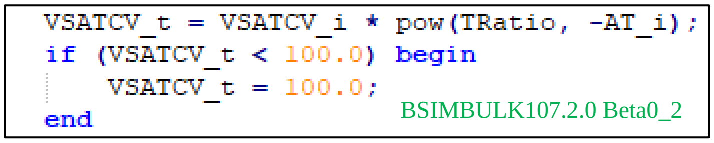
```
WARNING in file before.va, line 3048: bias-dependent $strobe() may degrade performance
```



```
    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

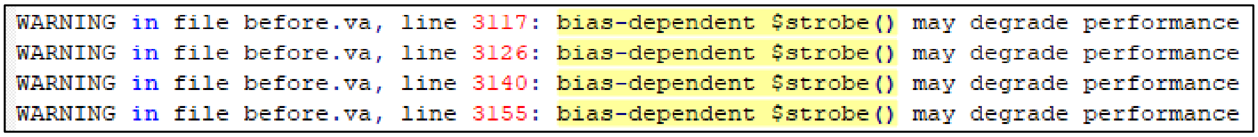
The $strobe statement was removed in this case as follows:



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

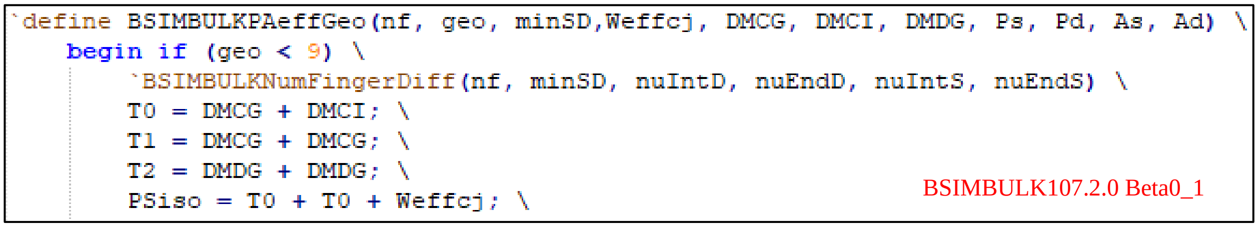
BSIMBULK107.2.0 Beta0_2

○



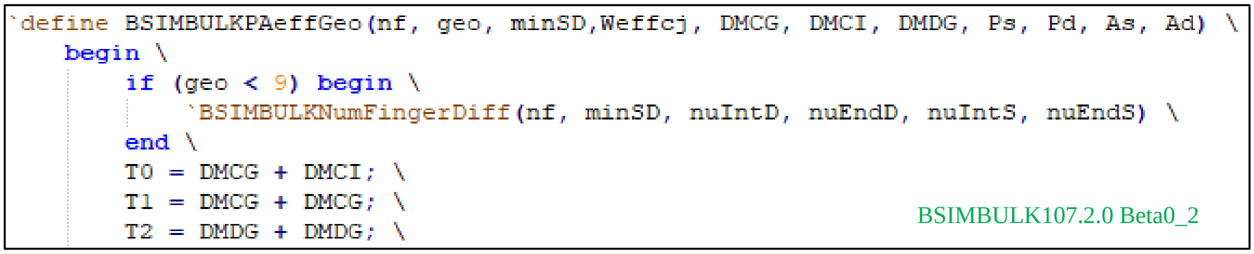
```
WARNING in file before.va, line 3117: bias-dependent $strobe() may degrade performance
WARNING in file before.va, line 3126: bias-dependent $strobe() may degrade performance
WARNING in file before.va, line 3140: bias-dependent $strobe() may degrade performance
WARNING in file before.va, line 3155: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

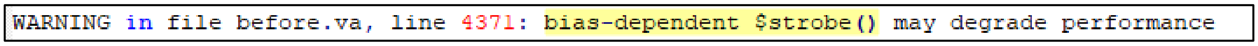
BSIMBULK107.2.0 Beta0_1



```
`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; \
```

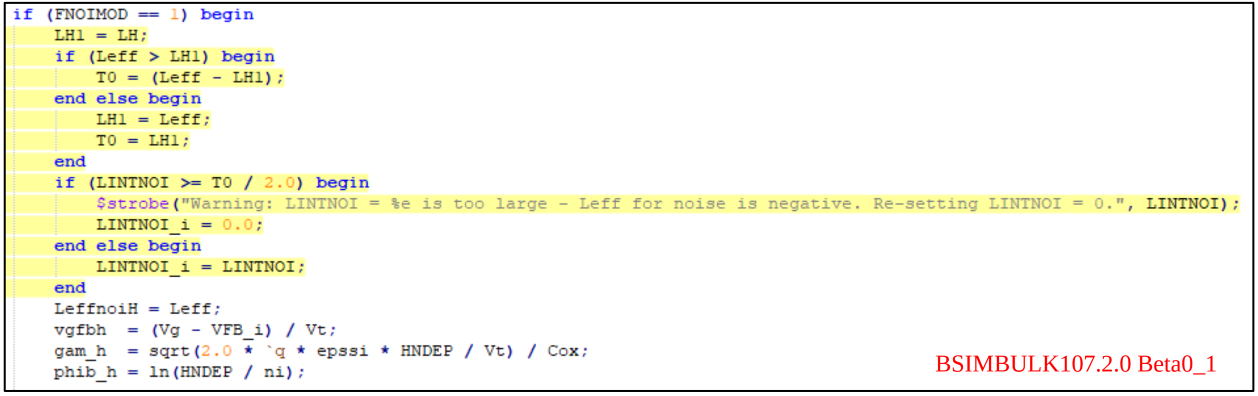
BSIMBULK107.2.0 Beta0_2

○



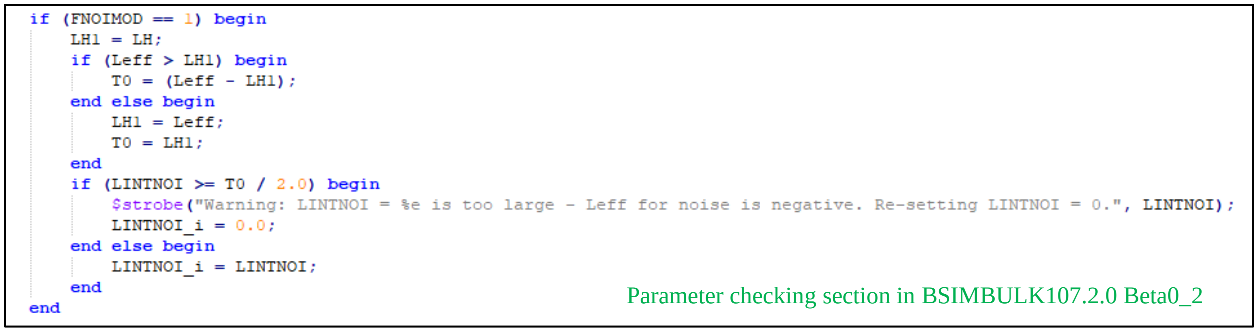
```
WARNING in file before.va, line 4371: bias-dependent $strobe() may degrade performance
```

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