Summary of Changes in BSIM-BULK107.0.0:

1. Kink due to large PSATB values.
2. Gm/Id tuning flexibility enhancement
3. Flexibility of tuning Cgg in strong inversion for Vds ≠0
4. Negative Cbg issue for CVMOD =0
5. Flexibility of tuning Cgg in accumulation
6. Add temperature dependence for the parameter EU
7. Changed the default value of the parameter "LH"
8. Valid condition for L <= LH?
9. Warning message for FNOIMOD=1
10. Enhance the BSIMBULK flicker noise tuning flexibility
11. Evaluating K2_EDGE and ETA0_EDGE independent of STI stress model
12. Device Type implementation in sub-surface leakage model.
13. Abnormal flicker noise performance when EDGEFET=1
14. Aligning BSIMBULK 1/f behavior with BSIM4 1/f behavior
15. Corrected implementation of sub-surface leakage model
16. Enhancing the tuning flexibility in the flicker noise model
17. Add flicker noise due to EDGEFET
18. Correction in description of parameters
19. Discrepancy in thermal noise model
20. Impact of non-zero C0 on thermal noise
21. Issue with the PSS (periodic state analysis)
22. Implementation of ABULK is updated

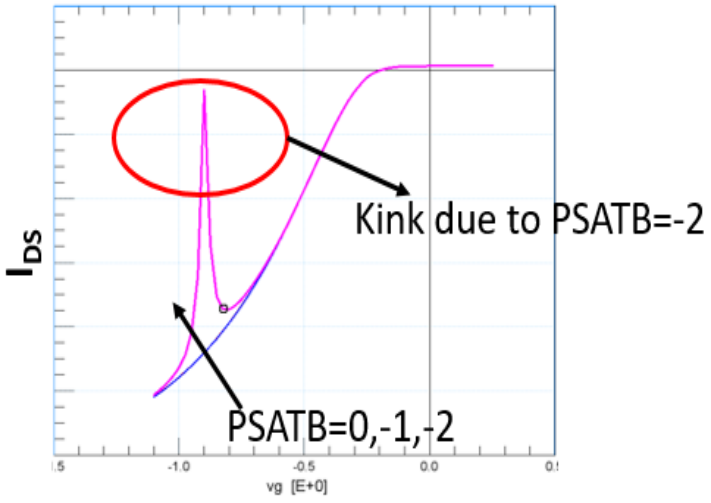
23. ‘phib’ and ‘gam’ are replaced by ‘phibCV’ and ‘gamCV’ respectively in CVMOD=1
24. NDEPEDGE and its binning
25. Description of TNOIMOD=1 in manual
26. Code Cleaning
27. Node collapsing and minimum value of Rdrain/Rsource
28. GDS Op-pt definition
29. Incorrect IGBBACC behavior in the accumulation region
30. Enhancement for High Voltage modeling
31. Warning message for U0 ≤ 0
32. Range of model parameters
33. Default value of NDEPEDGE
34. Kink in DC current due to PSATB
35. Rollback MNUD and MNUD1 from thermal noise model
36. Addressing IGBACC issue
37. Add GMIN across D-B and S-B junctions
38. Missing type in sub-surface leakage model
39. External/Internal node issue in sub-surface leakage model
40. HVMOD issue- CV model
41. Secondary impact ionization in HVMOD
42. Implementing ‘sigvds’ in flicker_noise

1. Kink due to large PSATB values.

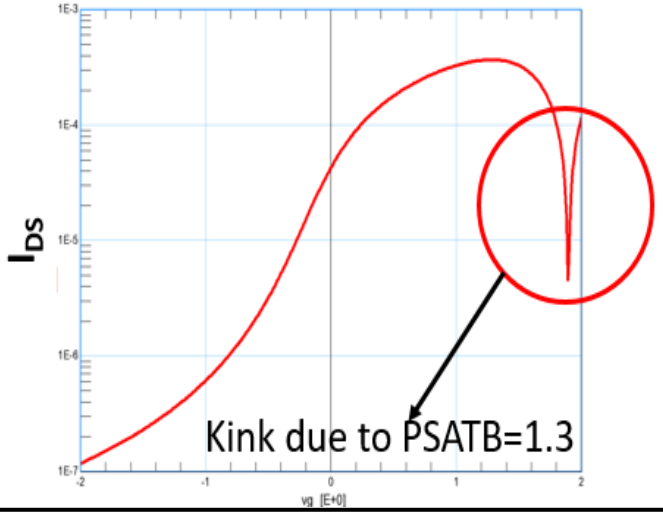
```
if (PSATB_i < 0.0) begin
    T1 = 1.0 / (1.0 + PSATB_i * Vbsx);
end else begin
    T1 = 1.0 - PSATB_i * Vbsx;
end
```

If PSATB_i * Vbsx approaches -1, so it may also give convergence issue.

Vds = -50mV, Vbs=-800mV, L=900nm, W=90nm

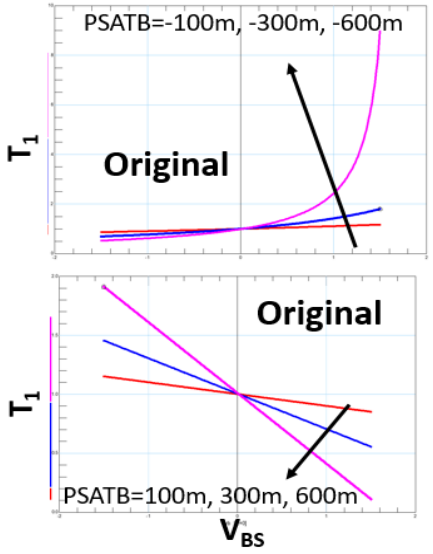


Vds = 50mV, Vb=1 V, L=900nm, W=90nm



Original implementation

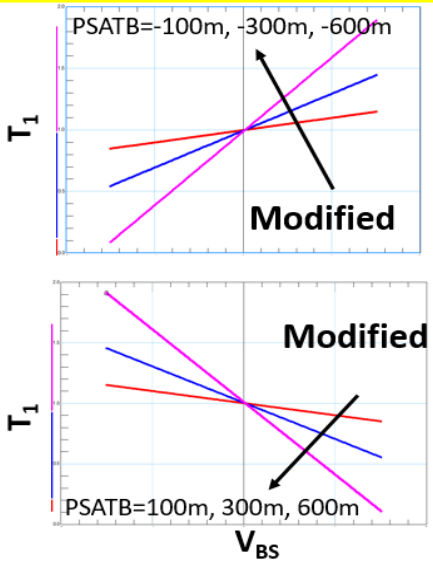
```
if (PSATB_i < 0.0) begin
    T1 = 1.0 / (1.0 + PSATB_i * Vbsx);
end else begin
    T1 = 1.0 - PSATB_i * Vbsx;
end
```



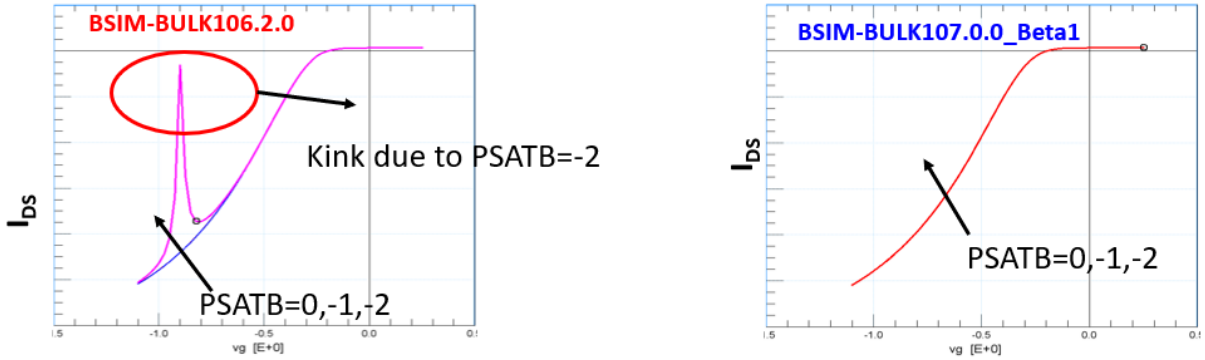
Unified Expression

```
T1 = hypsmooth((1.0 - PSATB_i * Vbsx), 1.0e-3);
```

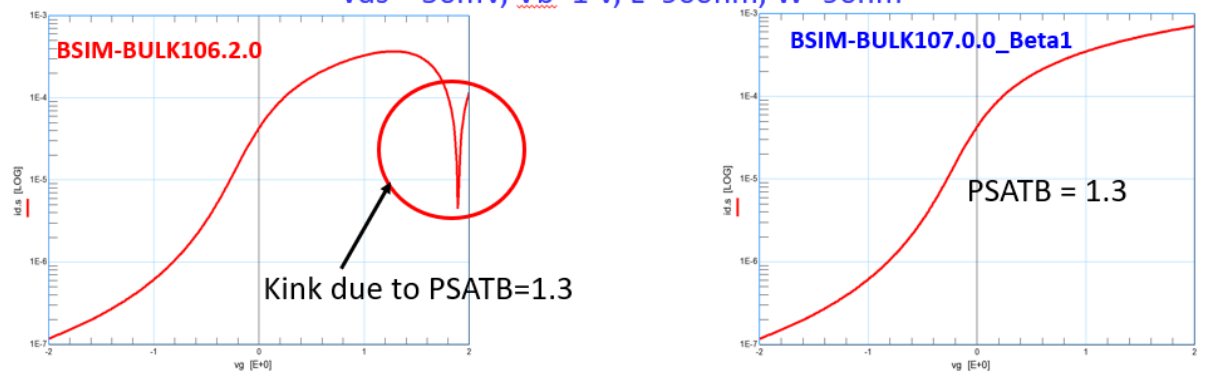
T1 is limited to small positive value.



Vds = -50mV, Vbs=-800mV, L=900nm, W=90nm



Vds = 50mV, Vb=1 V, L=900nm, W=90nm



2. Gm/Id Tuning Flexibility Enhancement

$$MNUD1 = \exp\left(-\frac{C0}{Max(0, COSI + COSISAT\,.\,(q_s - q_d)^2).(q_s + q_d) + 2.n_q.V_t}\right)$$

$$I_{DS} = I_{DS}.MNUD1$$

This model provides Gm/Id tuning flexibility in weak inversion region. There should be no effect in strong inversion.

In weak inversion: (COSI=1, COSISAT =1K, different values of C0)

In WI, q_s and q_d are very small

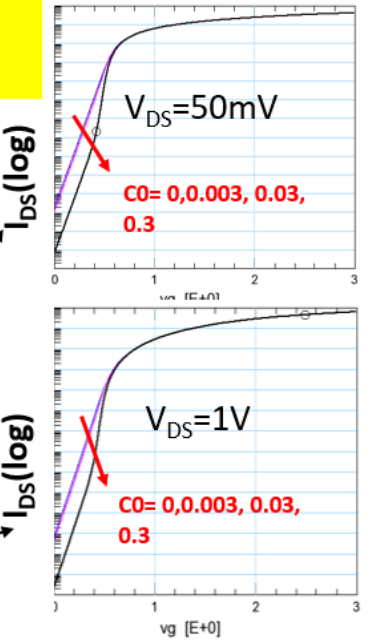
$$MNUD1 \sim \exp\left(-\frac{C0}{2.n_q.V_t}\right)$$

In strong inversion: (COSI=1, COSISAT =1K, different values of C0)

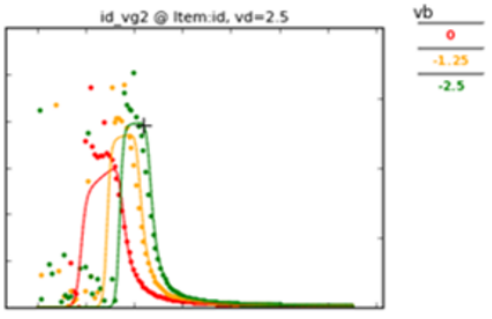
$$MNUD1 \sim \exp\left(-\frac{C0}{COSISAT\,.\,(q_s - q_d)^2.(q_s + q_d)}\right)$$

Mnud1 will be 1 Ids will not change in SI.

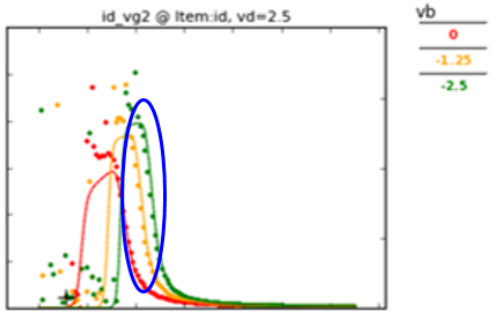
C0_t will provide tuning in gm/Id @low Vgs



C0=0, C0si=0.5, C0sisat=0



C0=-0.02, C0si=0.5, C0sisat=0



3. Flexibility of Tuning Cgg in Strong Inversion for Vds ≠0

```
VdsatCV = VdsatCV / ABULK;
// Normalized charge qdeff at drain end of channel
`Smooth(VdsatCV - Vs, 0.0, 1e-3, VdssatCV)
T7      = pow(Vds / VdssatCV , 1.0 / DELTA_t);
T8      = pow(1.0 + T7, -DELTA_t);
Vdseff = Vds * T8;
vdeff   = (Vdseff + Vs) * inv_Vt;
`BSIM_q(psip, phib, vdeff, gam, qdeff)
```

$$q_I = n_q \cdot \left[q_s + q_d + \frac{1}{3} \cdot \frac{ABULK^2 (q_s - q_d)^2}{1 + q_s + q_d} \right]$$

$$q_B = v_g - v_{fb} - \psi_p - (n_q - 1) \left[q_s + q_d + \frac{1}{3} \cdot \frac{ABULK^2 (q_s - q_d)^2}{1 + q_s + q_d} \right]$$

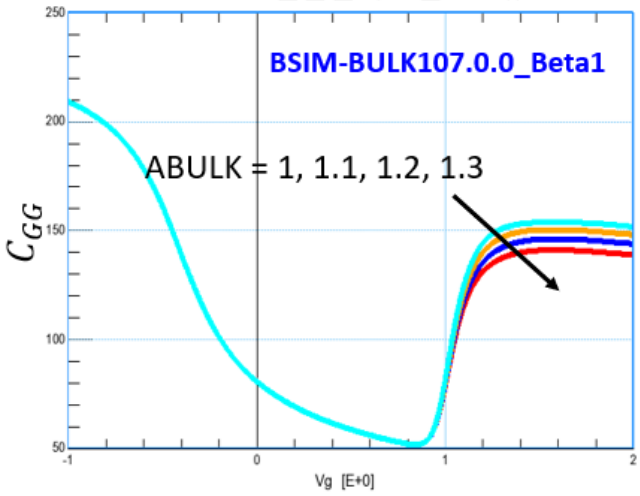
$$Q_d = \frac{n_q}{3} \left[q_s + 2.q_{deff} + \frac{1}{2} \cdot \left(1 + \frac{6}{5} q_s + \frac{4}{5} \cdot q_{deff} \right) \cdot ABULK^2 \cdot \left(\frac{q_s - q_{deff}}{1 + q_s + q_{deff}} \right)^2 \right]$$

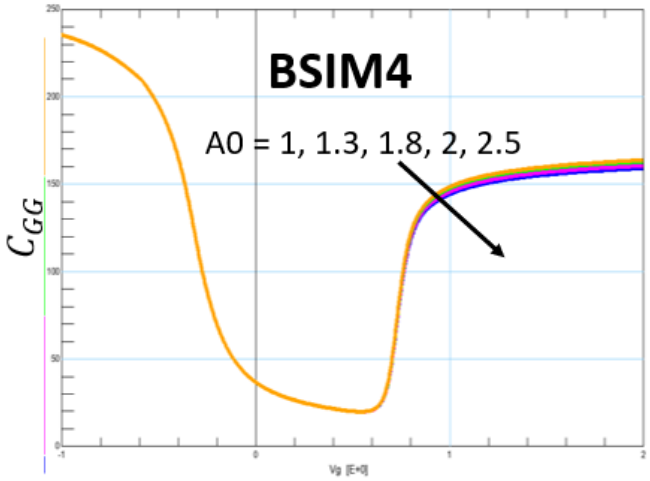
Default value of ABULK is 1

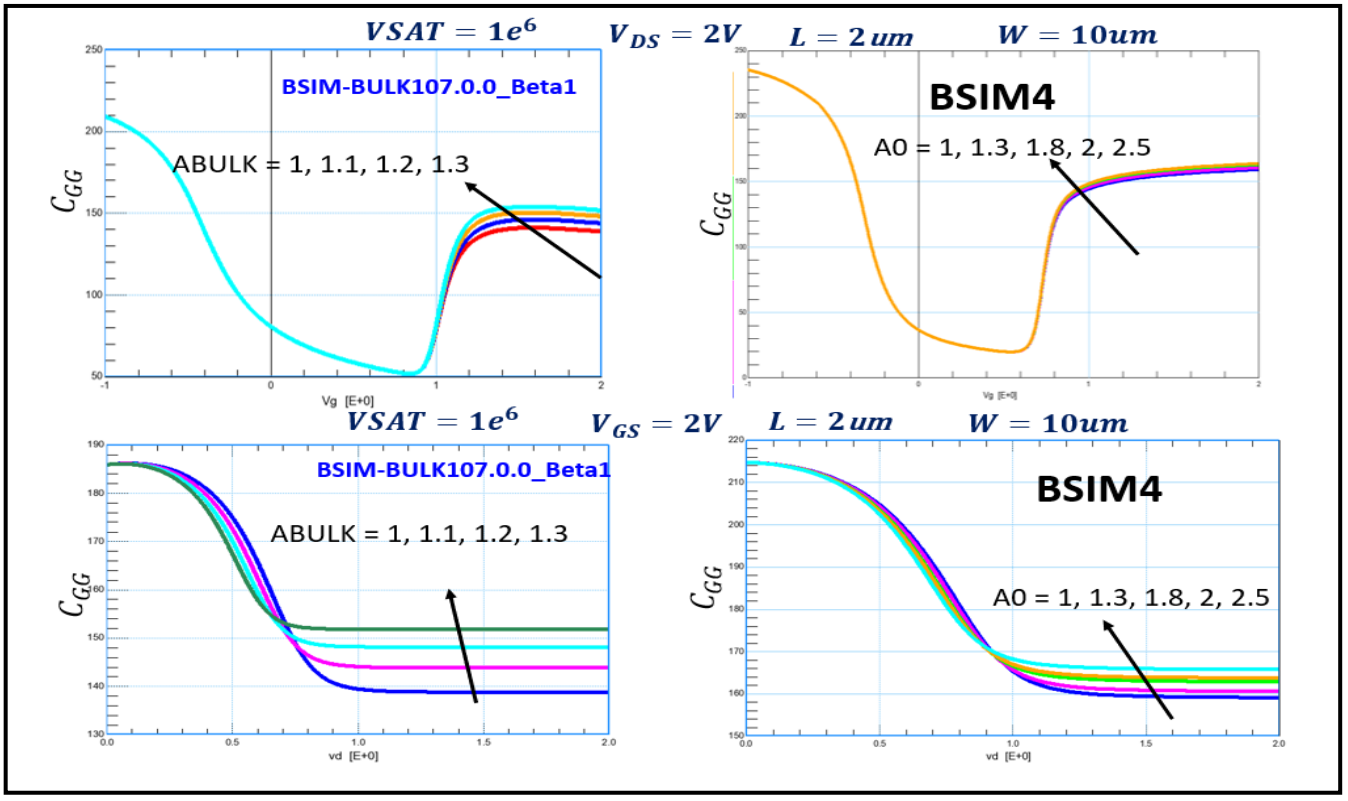
When Vds=0 → (qs-qdeff)=0 → **ABULK has no effect**

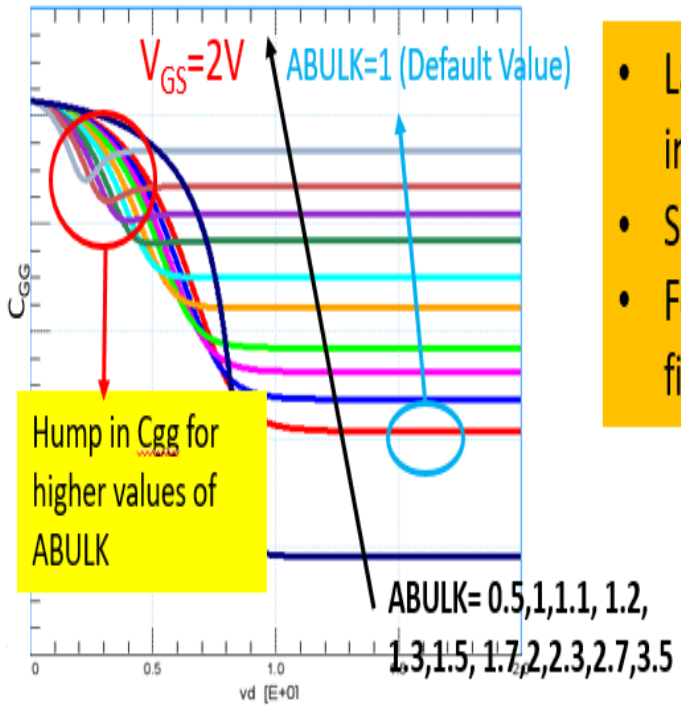
When Vds!=0 → (qs-qdeff)!=0 → **ABULK can tune Cgg**

$VSAT = 1e^6$ $V_{DS} = 2V$ $L = 2um$ $W = 10um$



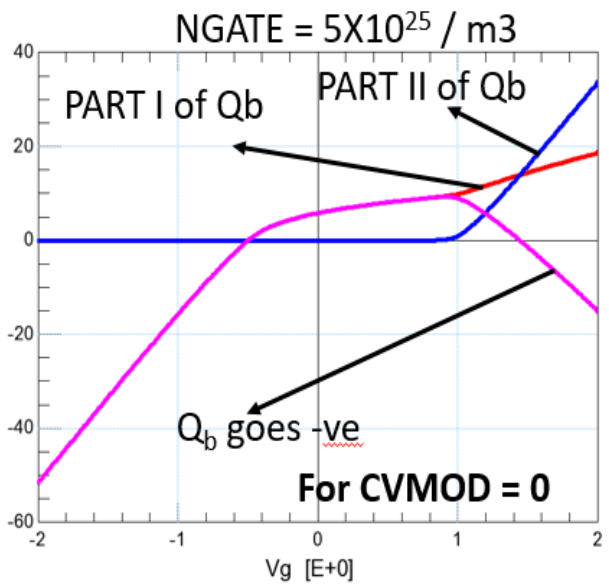


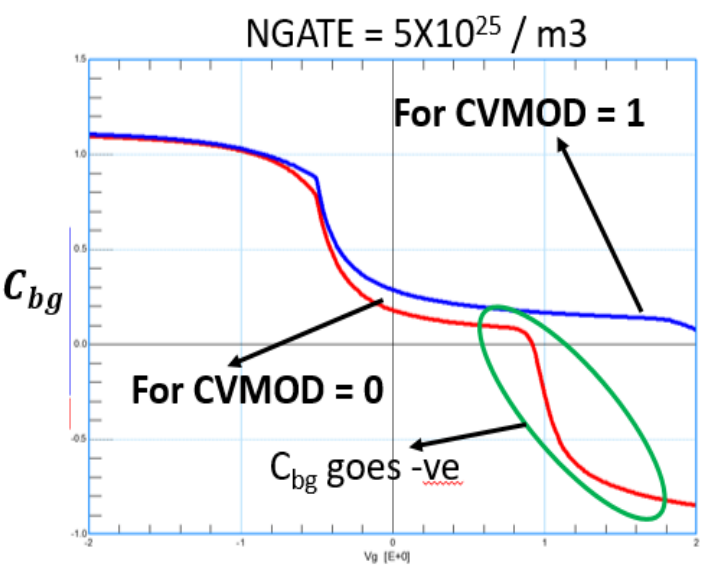




- Larger values of ABULK may cause humps in the capacitances.
- So we need to limit the ABULK value.
- For ABULK= 1 to 2 this model is working fine.

4. Negative Cbg issue for CVMOD =0






```
Qb = inv_MdL * (T1 + T2 + (T4*T7) - nq*( qs + qdeff + T8)) ;
```

PART I

This term contains the PDE through **invgamg2**

PART II

nq, qs,and qdeff terms don't contain the PDE

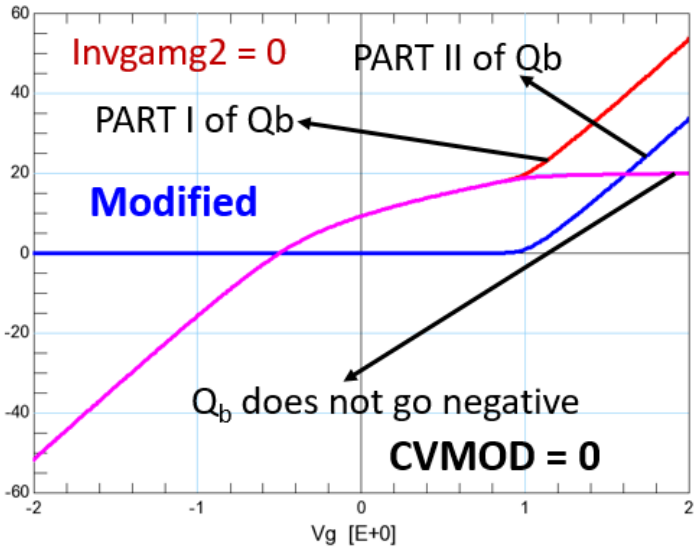
Whereas for CVMOD=1, both PART I and PART II contain the PDE

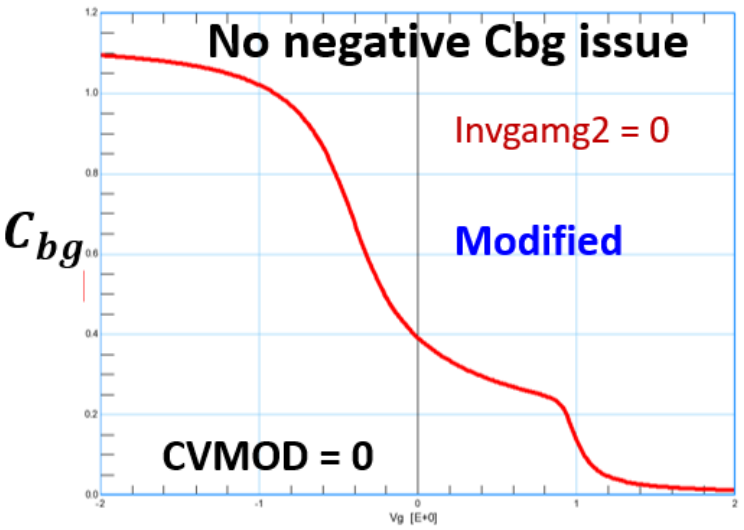
BSIM-BULK106.2.0

```
vgfbCV   = vgfb;
gamg2    = (2.0 * `q * epssi * NGATE_i) / (Cox * Cox * Vt);
invgamg2 = (NGATE_i > 0.0) ? (1.0 / gamg2) : 0.0;
if (CVMOD == 1) begin
    VFBCV_i = VFBCV_i + DELVTO;
```

BSIM-BULK107.0.0

```
vgfbCV   = vgfb;
gamg2    = (2.0 * `q * epssi * NGATE_i) / (Cox * Cox * Vt);
invgamg2 =0;
if (CVMOD == 1) begin
    VFBCV_i = VFBCV_i + DELVTO;
```





5. Flexibility of Tuning Cgg in Accumulation

BSIM-BULK106.2.0

`PO_psip(vgfb, gam, 0, phib_n, psip)

$$Q_B = v_g - v_{FB} - psip - (n_q - 1)\left[q_s + q_d + \frac{1}{3}\cdot\frac{(q_S - q_d)^2}{1 + q_s + q_d}\right]$$

BSIM-BULK107.0.0 (CVMOD = 1)

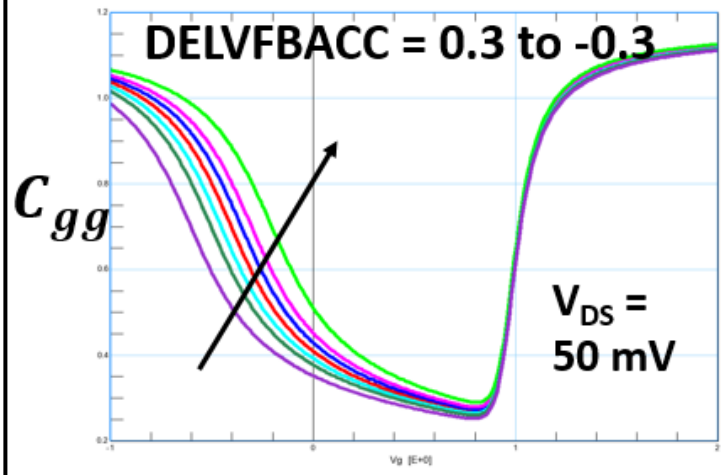
`PO_psip((vgfb + DELVFBACC * inv_nVt), gam, 0, phib_n, psipACC)

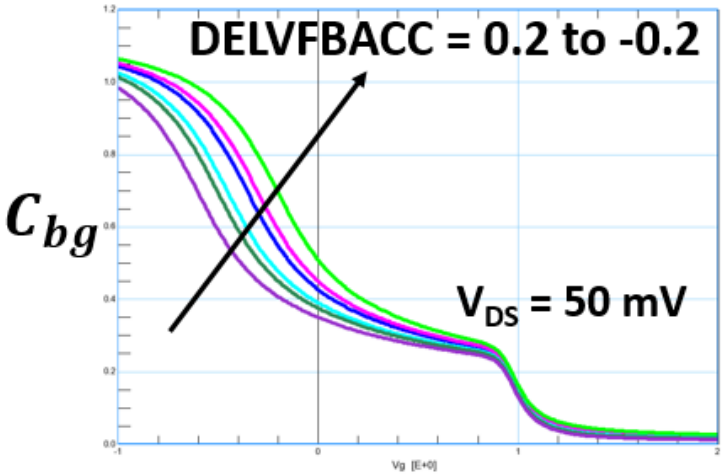
$$Q_B = v_g - v_{FB} - psipACC - (n_q - 1)\left[q_s + q_d + \frac{1}{3}\cdot\frac{(q_S - q_d)^2}{1 + q_s + q_d}\right]$$

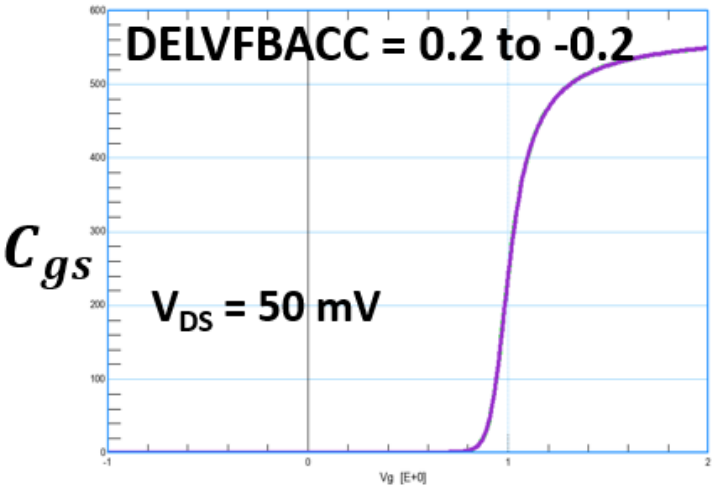
And **old psip** will be used for calculating the charge densities q_s and q_{deff}

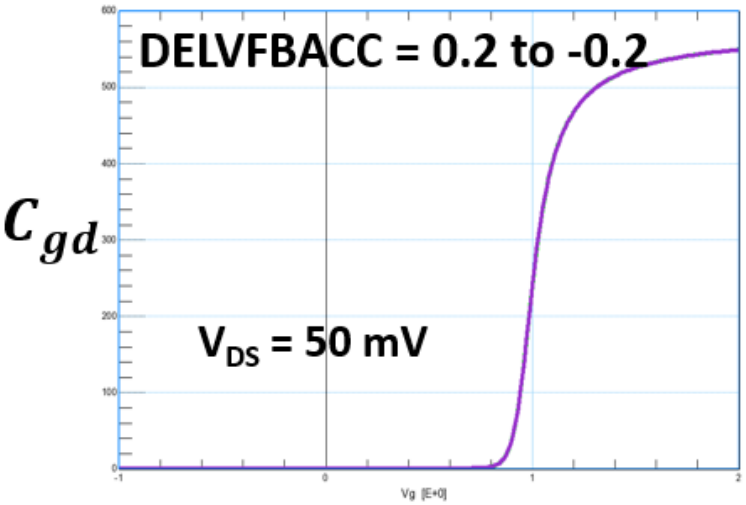
For CVMOD = 1

NGATE = $5X10^{26}$ / m3









6. Add Temperature Dependence for the parameter EU

BSIM-BULK106.2.0

```
// Ref: BSIM4 Model mobility model
T2 = pow(0.5 * (1.0 + (qis / qbs)), UCS_a);
T3 = (UA_a + UC_a * Vbsx) * pow(Eeffs, EU_i) + UD_a / T2;
T4 = 1.0 + T3;
```

BSIM-BULK107.0.0

```
// Ref: BSIM4 Model mobility model
T2 = pow(0.5 * (1.0 + (qis / qbs)), UCS_a);
T3 = (UA_a + UC_a * Vbsx) * pow(Eeffs, EU_t) + UD_a / T2;
T4 = 1.0 + T3;
```

7. Changed the default value of the parameter "LH"

BSIM-BULK106.2.0

```
`MPRoo( LH          ,30n                    ,"m"      ,0      ,L      ,"Length of Halo transistor" )
```

Upper limit for LH

```
Range error in device 'XCKT.Q1' : Parameter 'LH'
```

When L = 20 nm, LH = 30 nm (Default value), ADS gives an error

BSIM-BULK107.0.0

```
`MPRoo( LH          ,1.0e-8                 ,"m"      ,0.0    ,L      ,"Length of halo transistor" )
```

Default value of "LH" has changed to 10 nm.

Therefore, by using the modified definition, the same simulation gives no error.

8. Valid Condition for L <= LH?

FNOIMOD=1 and L > LH

$$S_{id}(Total) \cong \left[\frac{L_H.A}{L_H.A+(L-L_H).B}\right]^2.\frac{I_{DS}}{L_H^2}+\left[\frac{(L-L_H).B}{L_H.A+(L-L_H).B}\right]^2.\frac{I_{DS}}{(L-L_H)^2}$$

FNOIMOD=1 and L < LH

In the BSIM-BULK106.2.0

```
LeffnoiH = Leff;
if (Leff < LH) begin
    LeffnoiH = LH;
end
```

$$S_{id}(Total) \propto \frac{I_{DS}}{L_H^2} \quad \text{EQ1}$$

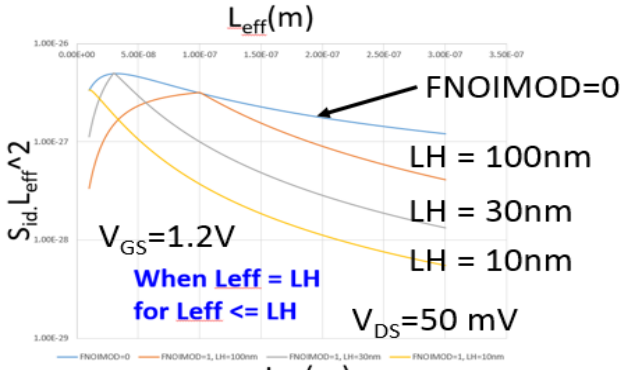
LH is fixed

FNOIMOD=0 (Uniformly doped device)

$$S_{id}(Total) \propto \frac{I_{DS}}{L_{eff}^2} \quad \text{EQ2}$$

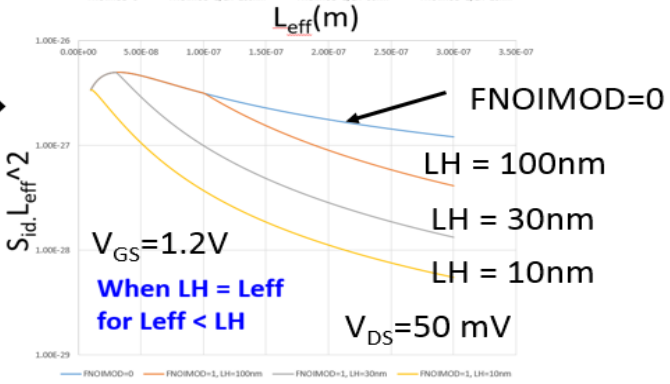
NOIA = NOIA2 = $1X10^{42}$; NDEP = HNDEP = $1X10^{23}/m^3$

For Leff < LH, the flicker noise of FNOIMOD=1 does not follow the noise of FNOIMOD=0, because EQ1 ≠ EQ2 (LH is fixed)




```
LeffnoiH = Leff;
if (Leff < LH) begin
    LH = LeffnoiH;
end
```

For Leff < LH, the flicker noise of FNOIMOD=1 follows the FNOIMOD=0 noise, because EQ1 = EQ2 (LH is scalable)



9. Warning message for FNOIMOD=1

BSIM-BULK106.2.0

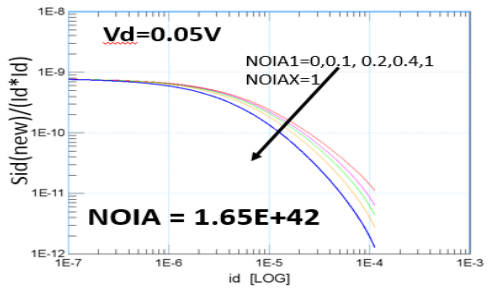
```
if(LINTNOI >= (Leff - LH) / 2.0) begin
    $strobe("Warning: LINTNOI = %e is too large - Leff for noise is negative. Re-setting LINTNOI = 0.", LINTN
    LINTNOI_i = 0.0;
end else begin
    LINTNOI_i = LINTNOI;
end
```

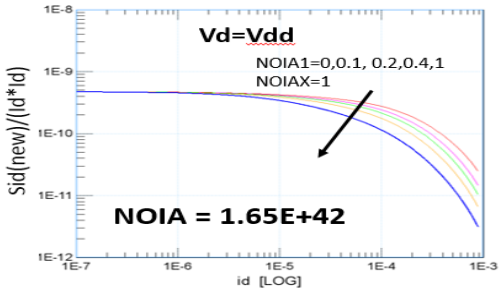
BSIM-BULK107.0.0

```
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
```

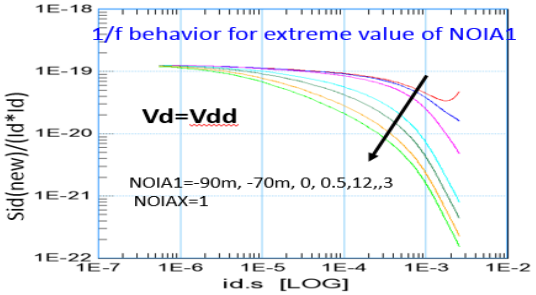
10. Enhance the BSIMBULK Flicker Noise Tuning Flexibility

$$Sid(new) = \frac{Sid(old)}{1 + NOIA1 * (qs - qdeff)}$$





- NOIA1 : fitting parameter
- Only affects strong region fitting
- NOIA1 should be positive as it is giving abnormal trends for negative values



11. Evaluating K2_EDGE and ETA0_EDGE independent of STI stress

Issue: if SA=0, SB=0 and SD=0 (default value), K2_EDGE and ETA0_EDGE (shown in red circle) can not get the updated values whenever there is a change in K2EDGE_i and ETA0EDGE_i.

```
if((SA > 0.0) && (SB > 0.0) && ((NF == 1.0) || ((NF > 1.0) && (SD > 0.0)))) begin
    --
    --
    if (EDGEFET == 1) begin
        vth0_stress_EDGE = (KVTH0EDGE_i / kstress_vth0) * (Inv_od - Inv_odref);
        k2_stress_EDGE   = (STK2EDGE_i / pow(kstress_vth0, LODK2)) * (Inv_od - Inv_odref);
        eta_stress_EDGE  = (STETA0EDGE_i / pow(kstress_vth0, LODETA0)) * (Inv_od - Inv_odref);
    end
    K2_EDGE   = K2EDGE_i + k2_stress_EDGE;
    ETA0_EDGE = ETA0EDGE_i + eta_stress_EDGE;
end else begin
    --
end
```

BSIM-BULK106.2.0

Solution: K2_EDGE and ETA0_EDGE is replaced by K2EDGE_i and ETA0EDGE_i .

```
if((SA >= 0.0) && (SB >= 0.0) && ((NF == 1.0) || ((NF > 1.0) && (SD >= 0.0)))) begin
    --
    --
    if (EDGEFET == 1) begin
        vth0_stress_EDGE = (KVTH0EDGE_i / kstress_vth0) * (Inv_od - Inv_odref);
        k2_stress_EDGE   = (STK2EDGE_i / pow(kstress_vth0, LODK2)) * (Inv_od - Inv_odref);
        eta_stress_EDGE  = (STETA0EDGE_i / pow(kstress_vth0, LODETA0)) * (Inv_od - Inv_odref);
    end
    K2EDGE_i   = K2EDGE_i + k2_stress_EDGE;
    ETA0EDGE_i = ETA0EDGE_i + eta_stress_EDGE;
end else begin
    --
end
```

BSIM-BULK107.0.0

12. Device Type Implementation in sub-surface leakage model

BSIM-BULK106.2.0

```
// Sub-Surface Leakage Drain Current
if (SSLMOD != 0) begin
    T1 = pow(NDEP_i / 1.0e23, SSLEXP1);
    T2 = pow(300.0 / DevTemp, SSLEXP2);
    SSL0_NT  = SSL0 * lexp(-T1 * T2);
    SSL1_NT  = SSL1 * T2 * T1;
    PHIB_SSL = SSL3 * tanh(lexp(devsign * SSL4 * (V(g, b) - VTH)));
    Issl     = sigvds * NF * Weff * SSL0_NT * lexp(-SSL1_NT * Leff) * lexp(PHIB_SSL / Vt) * (lexp(SSL2 * Vdsx / Vt) - 1.0);
    I(di, si) <+ Issl;
end
```

BSIM-BULK107.0.0

```
// Sub-Surface Leakage Drain Current
if (SSLMOD != 0) begin
    T1 = pow(NDEP_i / 1.0e23, SSLEXP1);
    T2 = pow(300.0 / DevTemp, SSLEXP2);
    SSL0_NT  = SSL0 * lexp(-T1 * T2);
    SSL1_NT  = SSL1 * T2 * T1;
    PHIB_SSL = SSL3 * tanh(lexp(devsign * SSL4 * (V(g, b) - VTH)));
    Issl     = sigvds * NF * Weff * SSL0_NT * lexp(-SSL1_NT * Leff) * lexp(PHIB_SSL / Vt) * (lexp(SSL2 * Vdsx / Vt) - 1.0);
    I(di, si) <+ devsign * Issl;
end
```

The device type is correctly handled using “devsign”.

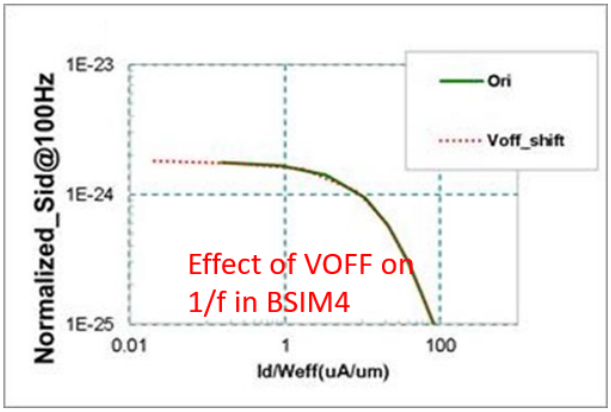
13. Abnormal Flicker Noise when EDGEFET=1

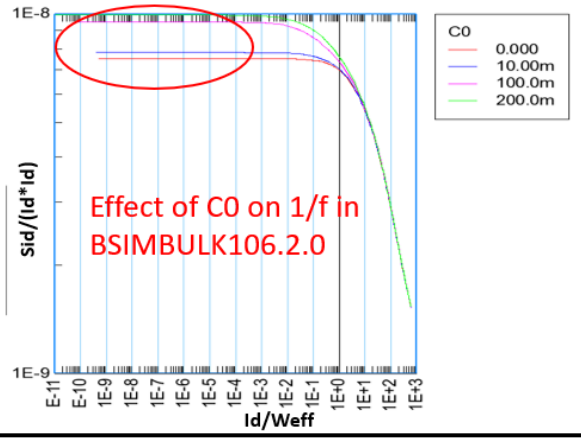
1/f model is shifted after EDGEFET model.

14. Aligning BSIMBULK 1/f behavior with BSIM4 1/f behavior

Issue:

- VOFF in BSIM4 does not affect 1/f noise. However, VOFF-like parameter, C0, in BSIMBULK significantly affect 1/f noise..
- TSMC wants that there should be no effect of C0 on 1/f noise





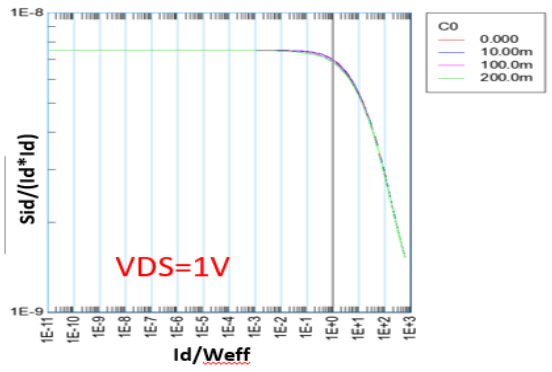
Solution:

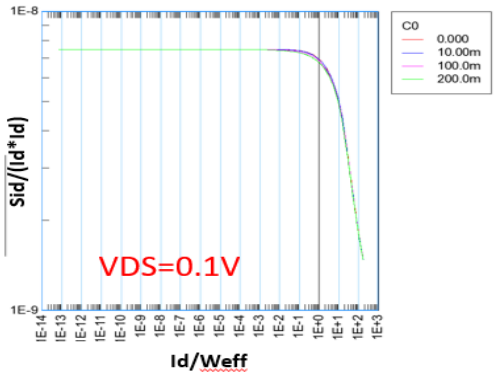
BSIM-BULK106.2.0

```
Nl  = 2.0 * nq * Cox * Vt * qdeff / `q;
N0  = 2.0 * nq * Cox * Vt * qs / `q;
```

BSIM-BULK107.0.0_Beta2

```
Nl  = 2.0 * nq * Cox * Vt * qdeff * (Mnud * Mnud1) / `q;
N0  = 2.0 * nq * Cox * Vt * qs * (Mnud * Mnud1) / `q;
```





Very small effect of non zero values of C0 on 1/f behavior

15. Corrected implementation of Sub-Surface Leakage model

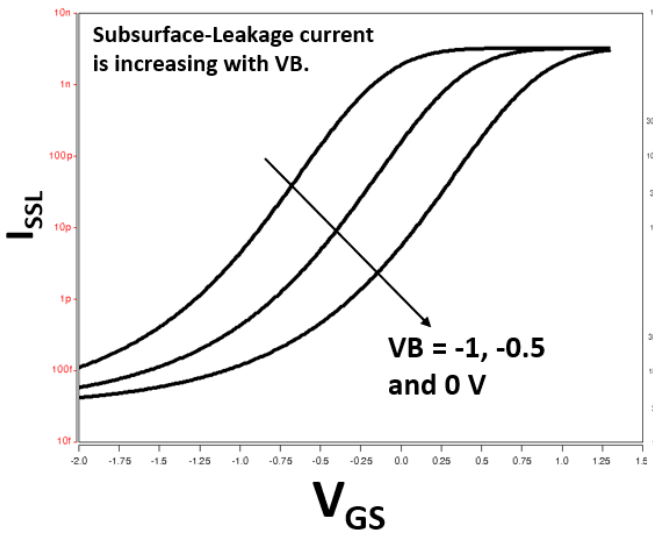
BSIM-BULK106.2.0

```
if (SSLMOD != 0) begin
    T1 = pow(NDEP_i / 1.0e23, SSLEXP1);
    T2 = pow(300.0 / DevTemp, SSLEXP2);
    SSL0_NT  = SSL0 * lexp(-T1 * T2);
    SSL1_NT  = SSL1 * T2 * T1;
    PHIB_SSL = SSL3 * tanh(lexp(devsign * SSL4 * (V(g, b) - VTH)));
    Issl     = sigvds * NF * Weff * SSL0_NT * lexp(-SSL1_NT * Leff) * lexp(PHIB_SSL / Vt) * (lexp(SSL2 * Vdsx / Vt) - 1.0);
    I(di, si) <+ Issl;
end
```

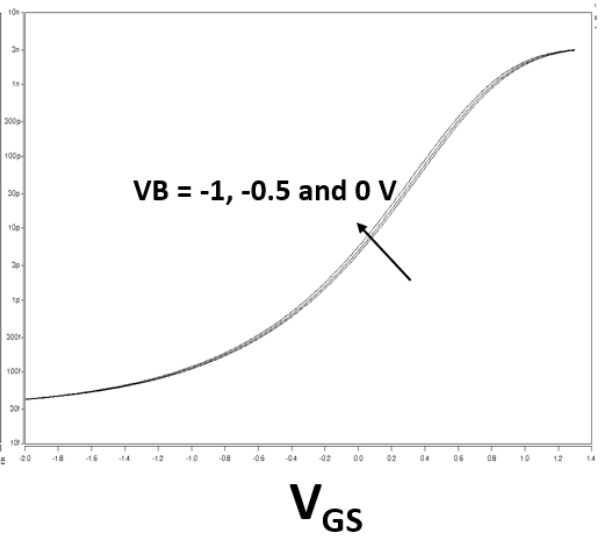
BSIM-BULK107.0.0

```
if (SSLMOD != 0) begin
    T1 = pow(NDEP_i / 1.0e23, SSLEXP1);
    T2 = pow(300.0 / DevTemp, SSLEXP2);
    T3 = ( devsign*SSL5 * V(bi,si)) / Vt;
    SSL0_NT  = SSL0 * lexp(-T1 * T2);
    SSL1_NT  = SSL1 * T2 * T1;
    PHIB_SSL = SSL3 * tanh(lexp(devsign * SSL4 * (V(gi, bi) - VTH - V(si,bi))));
    Issl     = sigvds * NF * Weff * SSL0_NT * lexp(T3) * lexp(-SSL1_NT * Leff) * lexp(PHIB_SSL / Vt) * (lexp(SSL2 * Vdsx / Vt) - 1.0);
    I(di, si) <+ devsign * Issl;
end
```

BSIM-BULK106.2.0

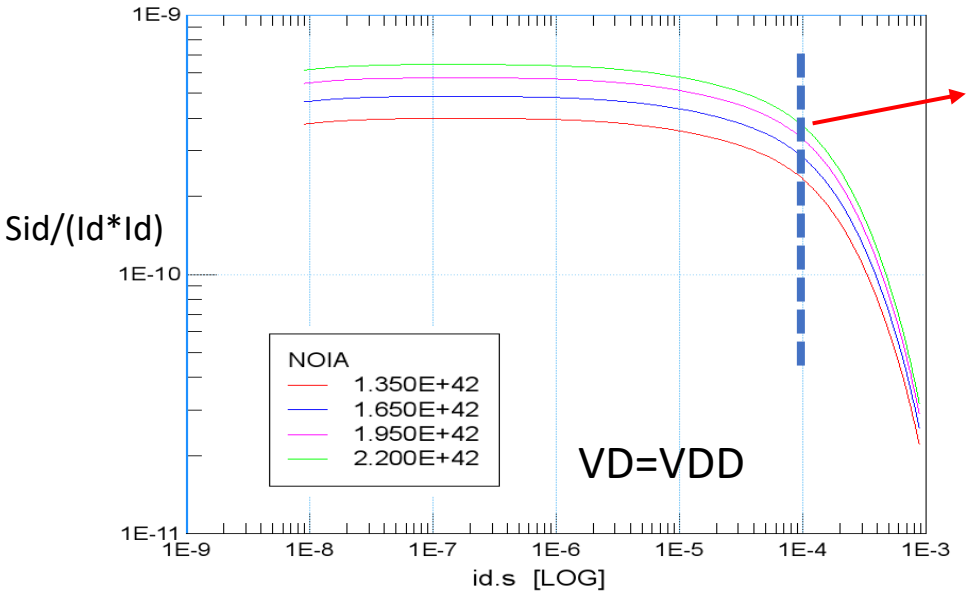


BSIM-BULK107.0.0



16. Enhancing the tuning flexibility in the flicker noise model.

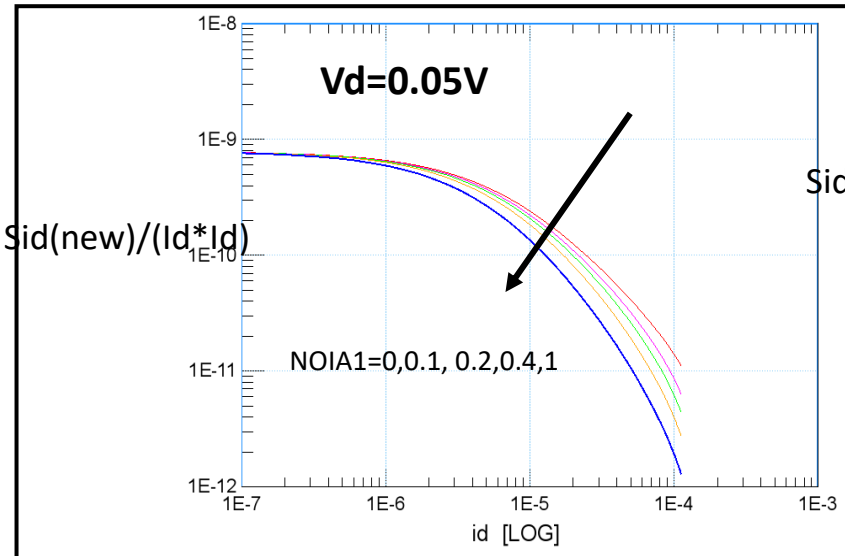
- In BSIM-BULK, the flicker noise fitting accuracy at the "turning point region" is not good enough.
- Turning point region is the region where device is going from on to off.

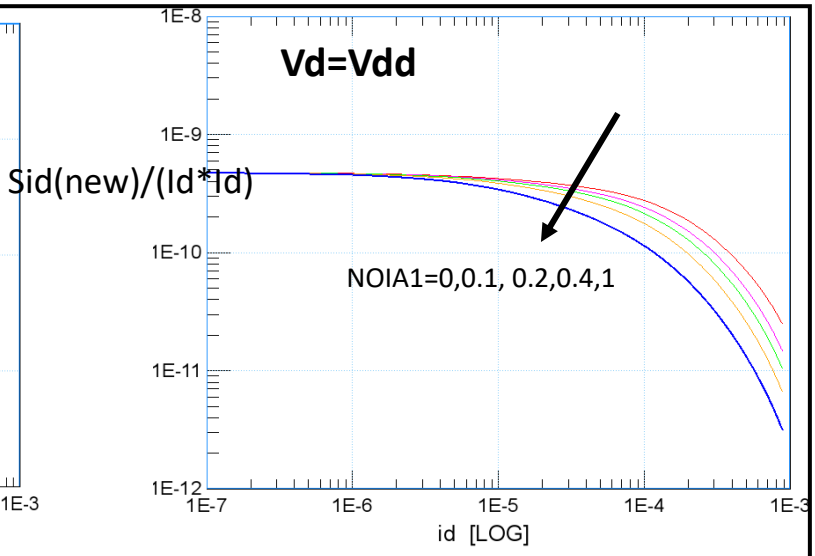


Insufficient parameters to tune the turning point.

Solution implemented in BSIM-BULK107.0.0_Beta1

$$Sid(new) = \frac{Sid(old)}{1 + NOIA1 * (qs - qdeff)}$$

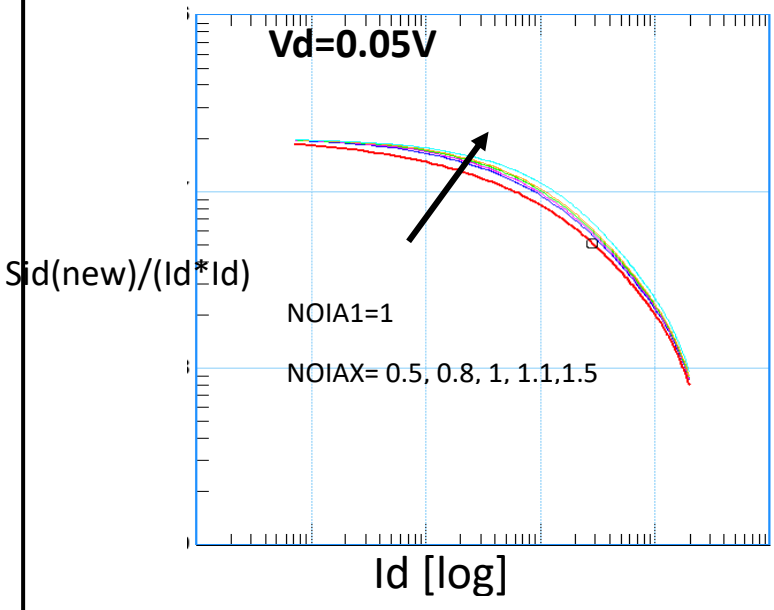


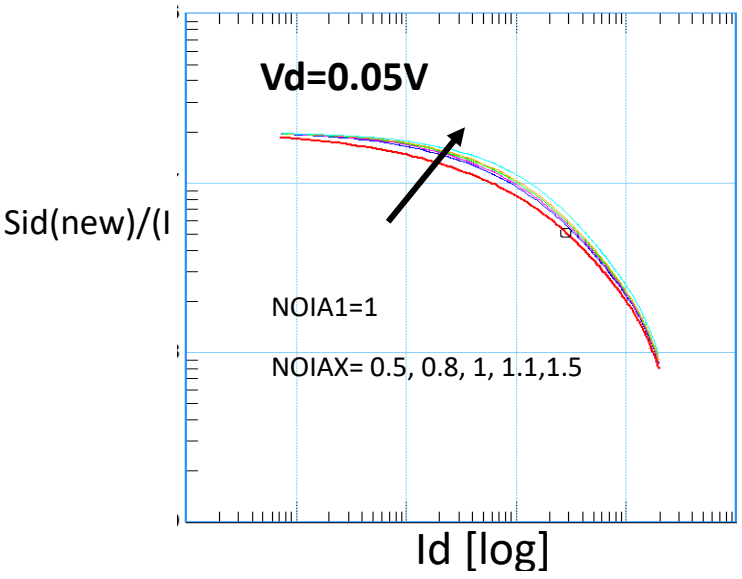


- The proposed solution only affects strong region fitting.
- NOIA1 should be positive.

Solution implemented in BSIM-BULK107.0.0:

$$Sid(new) = \frac{Sid(old)}{1 + NOIA1 * (qs - qdeff)^{NOIAX}}$$



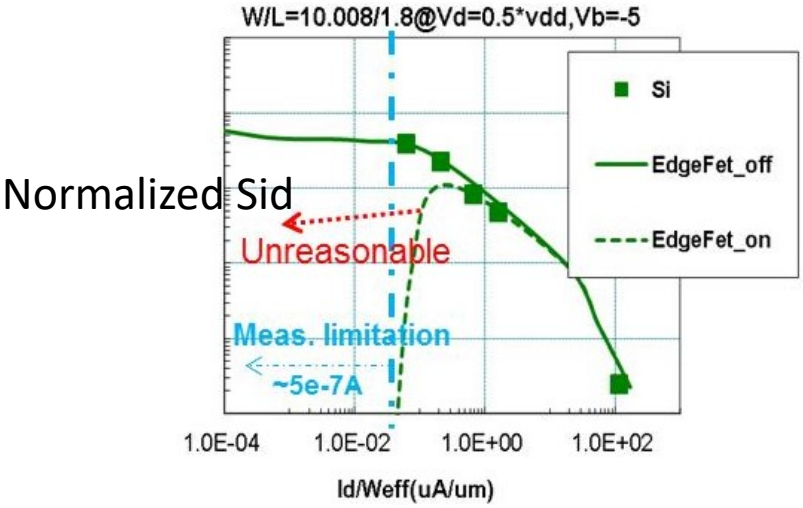


- NOIAX gives more fitting flexibility.

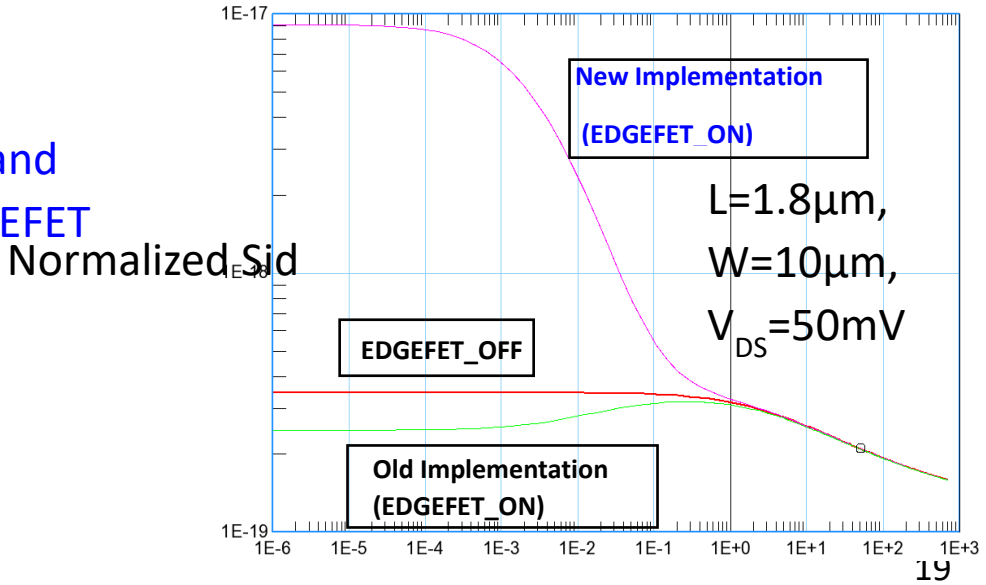
17. Add flicker noise due to EDGEFET

- When EDGEFET=1, there is an additional current component Ids_edge.
- The flicker noise model does not take into account the 1/f noise component from EDGEFET.

BSIM-BULK106.2.0



Solution: Calculate and add noise from EDGEFET with main noise.



18. Correction in description of parameters

BSIM-BULK106.2.0

```
`IPIcc( RGEOMOD   ,0     ,""   ,0   ,8   ,"Geometry-dependent source/drain
resistance,  0: RSH-based, 1: Holistic" )
```

BSIM-BULK107.0.0

```
`IPIcc( RGEOMOD   ,0     ,""   ,0   ,8   ,"Source/drain diffusion
resistance and contact model selector specifying the end S/D contact
type: point, wide or merged, and how S/D parasitics resistance is
computed" )
```

BSIM-BULK106.2.0

```
`IPRnb( SD ,0.0 ,"m" ,"Distance between neighbouring fingers" )
```

BSIM-BULK107.0.0

```
`IPRnb( SD ,0.0 ,"m" ,"Distance between neighboring fingers" )
```

BSIM-BULK106.2.0

```
`MPRnb( RNOIA ,0.577   ,"" ,"TNOIMOD = 1" )
`MPRnb( RNOIB ,0.5164  ,"" ,"TNOIMOD = 1" )
`MPRnb( RNOIC ,0.395   ,"" ,"TNOIMOD = 1" )
`MPRoo( TNOIA ,0.0     ,"" ,-inf ,inf ,"TNOIMOD = 1" )
`MPRoo( TNOIB ,0.0     ,"" ,-inf ,inf ,"TNOIMOD = 1" )
`MPRoo( TNOIC ,0.0     ,"" ,-inf ,inf ,"Correlation coefficient" )
```

BSIM-BULK107.0.0

```
`MPRnb( RNOIA ,0.577   ,"" ,"Noise parameter for TNOIMOD = 1" )
`MPRnb( RNOIB ,0.5164  ,"" ,"Noise parameter for TNOIMOD = 1" )
`MPRnb( RNOIC ,0.395   ,"" ,"Noise parameter for TNOIMOD = 1" )
`MPRoo( TNOIA ,0.0   ,"" ,-inf ,inf ,"Noise parameter for TNOIMOD = 1" )
`MPRoo( TNOIB ,0.0   ,"" ,-inf ,inf ,"Noise parameter for TNOIMOD = 1" )
`MPRoo( TNOIC ,0.0   ,"" ,-inf ,inf ,"Noise correlation coefficient
for TNOIMOD = 1" )
```

BSIM-BULK106.2.0

```
`IPRoo( SC ,0.0   ,"m" ,-inf   ,inf,"Distance to a single well edge if
<= 0.0, turn off WPE" )
```

BSIM-BULK107.0.0

```
`IPRoo( SC ,0.0   ,"m" ,-inf   ,inf,"Distance to a single well edge; if
<= 0.0, turn off WPE" )
```

19. Discrepancy in thermal noise model

BSIM-BULK106.2.0

In the VA code:

```
`MPRoo( TNOIA ,0.0  ,"" ,-inf ,inf ,"Noise parameter for TNOIMOD = 1" )
`MPRoo( TNOIB ,0.0  ,"" ,-inf ,inf ,"Noise parameter for TNOIMOD = 1" )
```

In the manual:

TNOIA	Coefficient of channel-length dependence of total channel thermal noise	1.5
TNOIB	Channel-length dependence parameter for channel thermal noise partitioning	3.5

BSIM-BULK107.0.0 (Backward incompatible change)

```
`MPRoo( TNOIA ,1.5  ,"" ,-inf ,inf ,"Noise parameter for TNOIMOD = 1" )
`MPRoo( TNOIB ,3.5  ,"" ,-inf ,inf ,"Noise parameter for TNOIMOD = 1" )
```

20. Impact of C0 on thermal noise

Issue:

- C0 significantly affect thermal noise behavior.
- TSMC wants there should be no effect of non zero C0 on thermal noise.

Proposed Solution:

- Considering MNUD and MNUD1 parameters with qs and qdeff

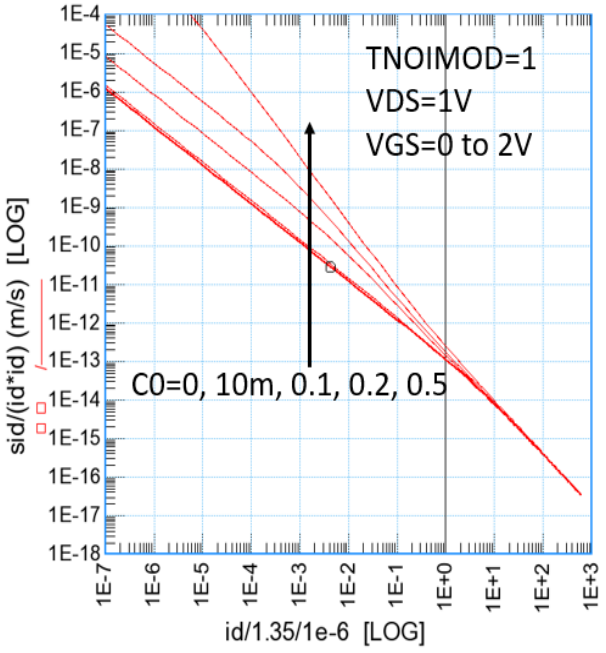
- **BSIM-BULK106.2.0**:

  Ids=k1*(qs+qdeff)*MNUD*MNUD1;
  Sid=K2*(qs+qdeff);

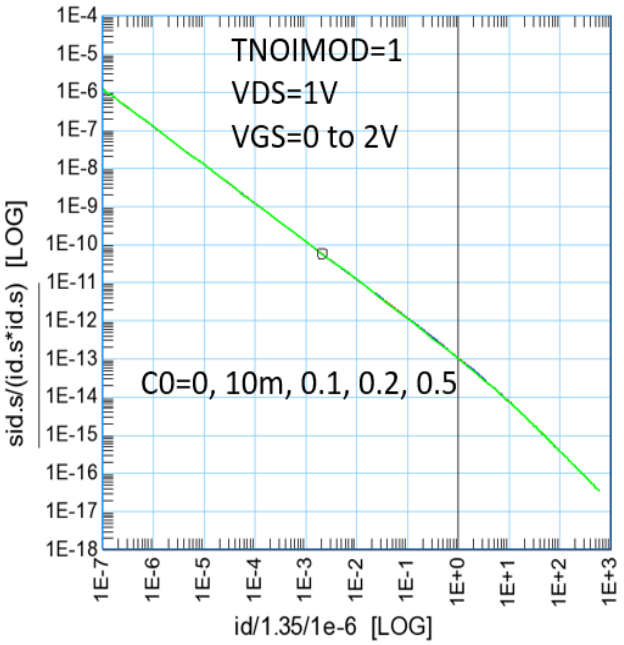
- **BSIM-BULK107.0.0_Beta3**:

  Ids=k1*(qs+qdeff)*MNUD*MNUD1;
  Sid=K2*(qs+qdeff)*MUND*MNUD1;

BSIM-BULK106.2.0



BSIM-BULK107.0.0_Beta3



21. Issue with the PSS (periodic state analysis)

Issue:

- Error found by SPECTRE during periodic steady state analysis `pss1'.
- Reason: 'Esatnoi' and 'Nt' parameters are used before they are set.

Proposed Solution:

- 'Esatnoi' and 'Nt' parameters are set before they are used.

22. Implementation of ABULK is updated

- 'ABULK' is an empirical approach for tuning CGG at non zero VDS.
- Good to keep the value of 'ABULK' near to 1 (1 to 1.25 approx.)
- ABULK implementation:

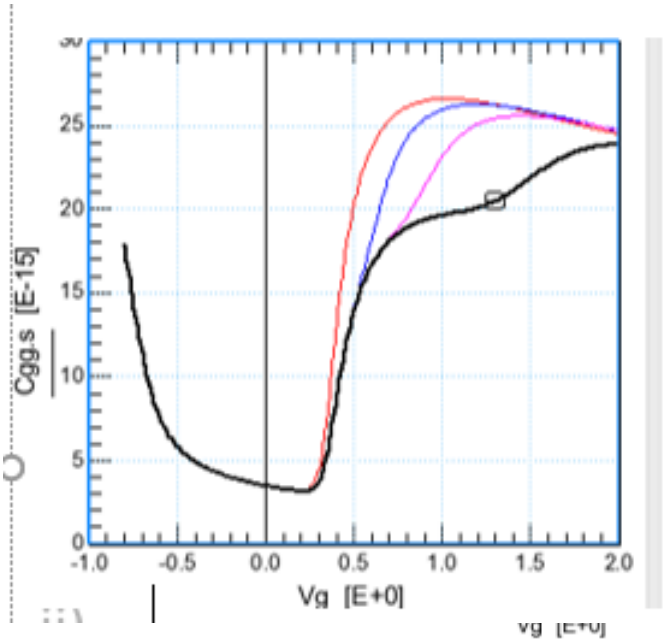
BSIM-BULK107.0.0_Beta1

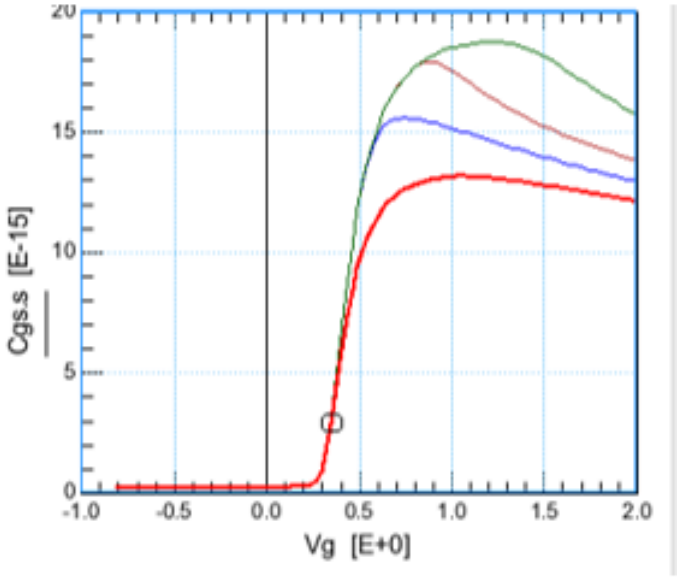
```
VdsatCV    = VdsatCV / ABULK;
DQSD  = (qs - qdeff) * ABULK;
DQSD2 = (qs - qdeff) * (qs - qdeff)
              * (ABULK * ABULK);
```

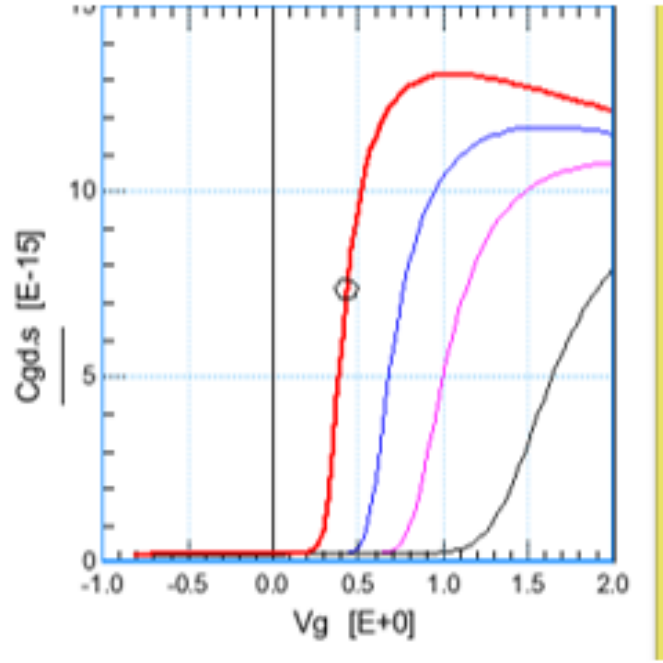
BSIM-BULK107.0.0

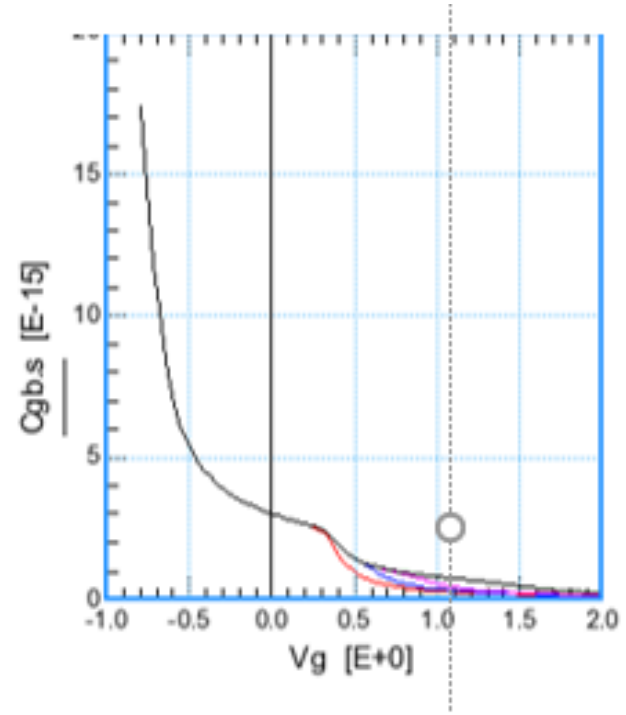
```
`Smooth(VdsatCV - Vs, 0.0, 1e-3, VdssatCV)
VdssatCV    = VdssatCV / ABULK;
T5 = (ABULK * Dvsat * inv_MdL) / (1.0 + qs + qdeff);
```

Different capacitance w.r.t VGS (VDS= 0, 0.2V, 0.4V, 0.8V,

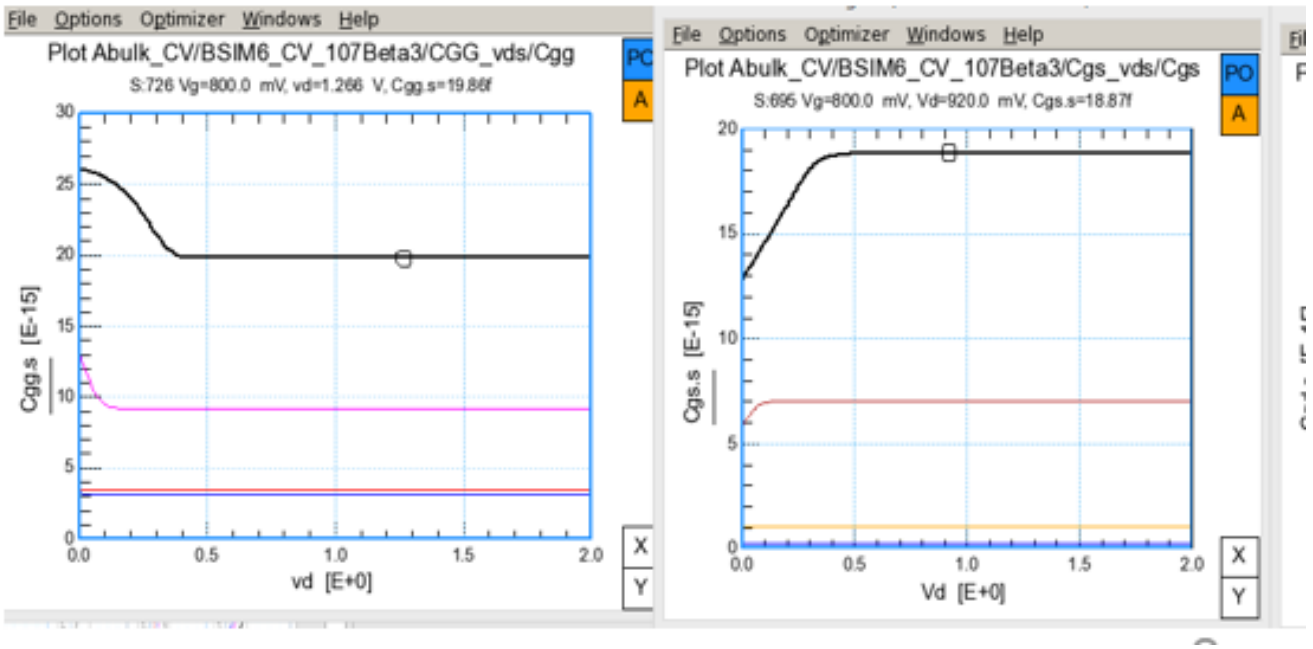


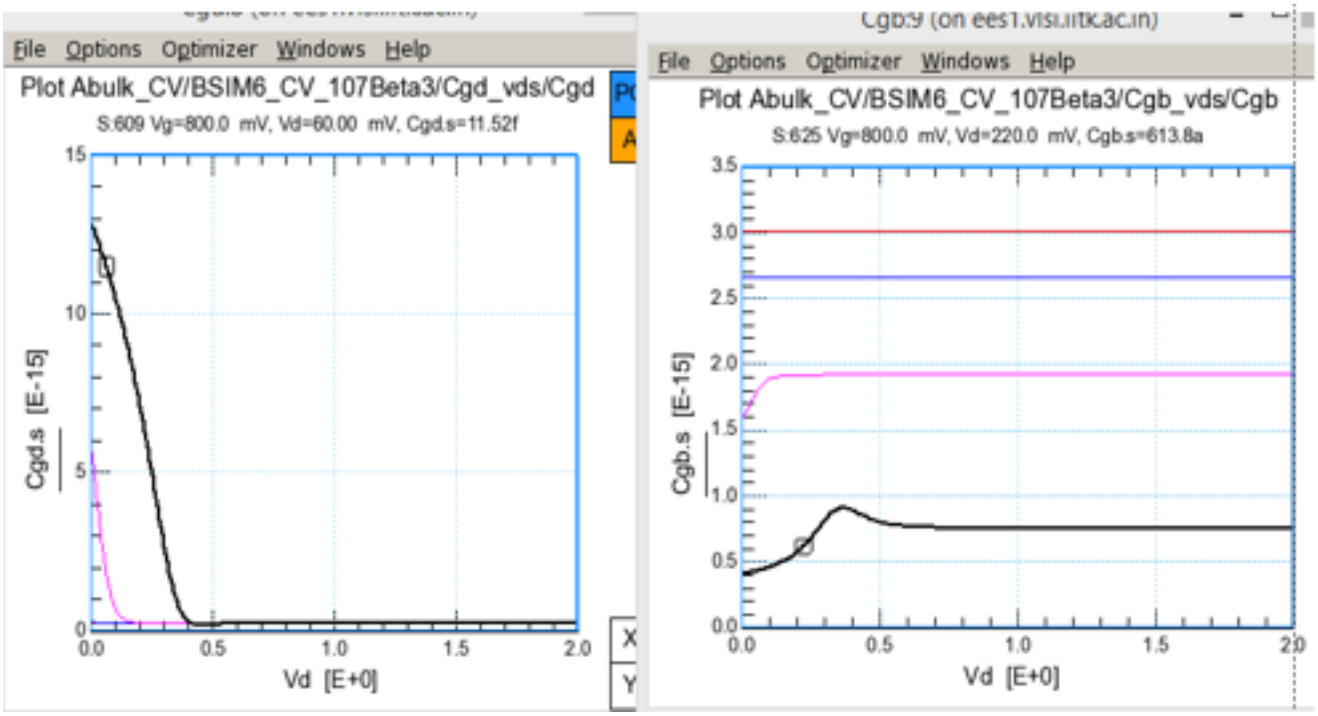






Different capacitance w.r.t VDS (VGS= 0, 0.2V, 0.4V, 0.8V, ABULK=1.25)



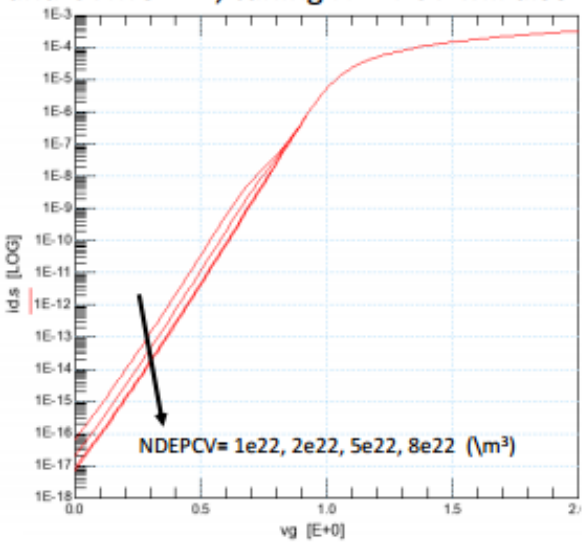


23. "phib" and "gam" are replaced by "phibCV" and "gamCV" respectively in CVMOD=1

Issue:

- NDEPCV is supposed to only affect the CV model, when CVMOD=1
- However, we find that when both EDGEFET=1 and CVMOD=1, tuning NDEPCV will also affect the IV simulation results.

- When CVMOD=1
  phib = lln(**NDEPCV_i** / ni)
- When EDGEFET=1,
  phib_n = phib / n
- It means NDEPCV will also be used EDGEFET, which will affect IV results.



Solution:

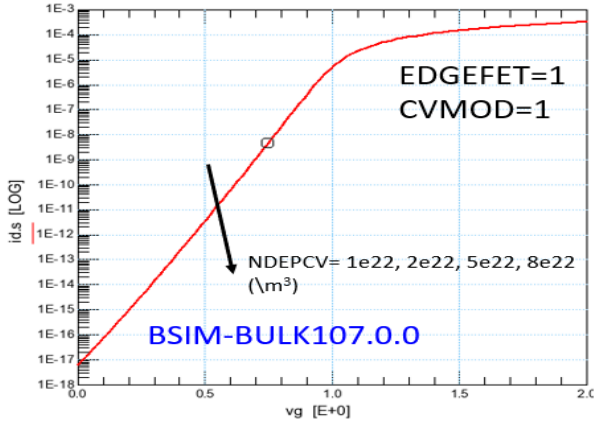
- Rename common variable used in CVMOD=1 and EDGEFET=1
- phibCV instead of phib; gamCV instead of gam

BSIM-BULK106.2.0

```
if (CVMOD == 1) begin
  phib  = lln(NDEPCV_i / ni);
  gam   = sqrt(2.0 * `q * epssi * NDEPCV_i * inv_Vt) / Cox;
end
```

BSIM-BULK107.0.0

```
if (CVMOD == 1) begin
  phibCV = lln(NDEPCV_i / ni);
  gamCV  = sqrt(2.0 * `q * epssi * NDEPCV_i * inv_Vt) / Cox;
end
```



24. NDEPEDGE and its binning

```
`MPRnb( NDEPEDGE  ,1e24   ,"1/m^3"  ,"Channel doping concentration for I-V" )
`MPRnb( LNDEPEDGE ,0.0    ,"1/m^2"  ,"Length dependence of NDEP" )
`MPRnb( WNDEPEDGE ,0.0    ,"1/m^2"  ,"Width dependence of NDEP" )
`MPRnb( PNDEPEDGE ,0.0    ,"1/m"    ,"Area dependence of NDEP " )

NDEPEDGE_i        = NDEPEDGE + BIN_L * LNDEPEDGE + BIN_W * WNDEPEDGE + BIN_WL * PNDEPEDGE;
```

NDEPEDGE_i is used in place of NDEP_i in all subsequent calculations in EDGEFET module.

```
phib_edge      = lln(NDEPEDGE_i / ni);
Vbi_edge       = lln(NDEPEDGE_i * NSD_i / (ni * ni));
T1DEP          = sqrt(2.0 * epssi / (`q * NDEPEDGE_i));
gam_edge       = sqrt(2.0 * `q * epssi * NDEPEDGE_i * inv_nVt) / Cox;
```

25. Description of TNOIMOD=1 in manual

Noise models in BSIM- BULK	Origin
Thermal Noise (TNOIMOD=0)	BSIM4 (TNOIMOD=0)
Thermal Noise (TNOIMOD=1)	BSIM4 (TNOIMOD=2)

(BSIM4 (TNOIMOD=1) defined as holistic model is not used in BSIM-BULK)

BSIM-BULK106.2.0(in manual) :

TNOIMOD = 1 is defined as Holistic Model

Updated Manual:

TNOIMOD = 1 is Correlated Noise Model (it is equivalent to TNOIMOD=2 of BSIM4)

26. Code Cleaning

BSIM-BULK106.2.0

```
if ((SHMOD != 0) && (RTH0 > 0.0) && (Weff_SH > 0.0)) begin
    T0   = lln(NDEP_i * NSD_i / (ni * ni));
    Vbi  = sqrt(T0 * T0 + 1.0e-6);
end else begin
    Vbi  = lln(NDEP_i * NSD_i / (ni * ni));
end
.
.
Vbi_edge       = lln(NDEPEDGE_i * NSD_i / (ni * ni));
.
.
dvth_sce   = theta_sce_edge * (Vbi - Phist);
```

Redundant

BSIM-BULK107.0.0

```
if ((SHMOD != 0) && (RTH0 > 0.0) && (Weff_SH > 0.0)) begin
        T0   = lln(NDEPEDGE_i * NSD_i / (ni * ni));
        Vbi_edge = sqrt(T0 * T0 + 1.0e-6);
    end else begin
        Vbi_edge = lln(NDEPEDGE_i * NSD_i / (ni * ni));
    end
.
.
.
dvth_sce   = theta_sce_edge * (Vbi_edge - Phist);
```

BSIM-BULK106.2.0

```
`MPRnb( U0R   ,U0     ,"m2/V/s"      ,"Reverse-mode Low Field mobility." )
```

BSIM-BULK107.0.0

```
`MPRnb( U0R   ,U0     ,"m^2/V/s"     ,"Reverse-mode Low Field mobility." )
```

27. Node collapsing and minimum value of Rdrain/Rsource

There are two items related with this issue:

i) Minimum value of Rdrain/ Rsource
ii) Node Collapsing issue in RDSMOD=0 (RDSMOD=0 has bias independent external source/drain resistance)

- RSourceGeo/RDrainGeo always has minimum value of 1mΩ. It means that we can never eliminate source/drain resistance.

- We introduce a new model parameter **"minr".** It is the value below which the simulator expects elimination of source/drain resistance and it will improve simulation efficiency without significantly altering the results.

```
`MPRoz( minr          ,$simparam("minr", 1m) ,"Ohm"     ,"")
```

BSIM-BULK106.2.0

```
// Clamping of S/D resistances
if (RSourceGeo <= 1.0e-3) begin
    RSourceGeo = 1.0e-3;
end
if (RDrainGeo <= 1.0e-3) begin
    RDrainGeo = 1.0e-3;
end
```

BSIM-BULK107.0.0

```
if (RDSMOD == 0) begin
    if (RSourceGeo < minres) begin
        RSourceGeo = 0;
    end
    if (RDrainGeo < minres) begin
        RDrainGeo = 0;
    end
end else begin
    if (RSourceGeo <= minres) begin
        RSourceGeo = minres;
    end
    if (RDrainGeo <= minres) begin
        RDrainGeo = minres;
    end
end
```

RDSMOD = 0:
Eliminate resistance when
Rdraingeo/Rsource geo < minres

RDSMOD ! = 0 :
Clamping resistance when
Rdraingeo/Rsource geo < minres

BSIM-BULK106.2.0

```
if (RDSMOD != 2) begin
    gdpr = 1.0 / Rdrain;    // Note: gdpr consic
    gspr = 1.0 / Rsource;   // Note: gspr consic
    I(d, di) <+ V(d, di) * gdpr;
    I(s, si) <+ V(s, si) * gspr;
    I(d, di) <+ white_noise(Nt * gdpr, "rd");
    I(s, si) <+ white_noise(Nt * gspr, "rs");
end else begin
    V(d, di) <+ 0.0;
    V(s, si) <+ 0.0;
end
```

BSIM-BULK107.0.0

```
if (RDSMOD !=2 && RDrainGeo >0) begin
    gdpr = 1.0 / Rdrain;    // Note: gdpr consi
    I(d, di) <+ V(d, di) * gdpr;
    I(d, di) <+ white_noise(Nt * gdpr, "rd");
end else begin
    V(d, di) <+ 0.0;
end

if (RDSMOD !=2 && RSourceGeo >0) begin
    gspr = 1.0 / Rsource;    // Note: gspr cons
    I(s, si) <+ V(s, si) * gspr;
    I(s, si) <+ white_noise(Nt * gspr, "rs");
end else begin
    V(s, si) <+ 0.0;
end
```

BSIM-BULK106.2.0

```
if ((SHMOD != 0) && (RTH0 > 0.0)) begin
    if (RDSMOD != 2) begin
        Pwr(t) <+ -(devsign * sigvds * ids * V(di, si) + V(d, di) * V(d, di) / Rdrain + V(s, si) * V(s, si) / Rsource) + ddt(delTemp1
         * cth) + delTemp1 * gth;
    end else begin
        Pwr(t) <+ -(devsign * sigvds * ids * V(di, si))  + ddt(delTemp1 * cth) + delTemp1 * gth;
    end
end else begin
    Temp(t) <+ 0.0;
end
```

BSIM-BULK107.0.0

```
if ((SHMOD != 0) && (RTH0 > 0.0)) begin
    Pdiss = devsign * sigvds * ids * V(di, si);
    if (RDSMOD !=2 && RDrainGeo >0) begin
        Pdiss = Pdiss + V(d, di) * V(d, di) / Rdrain;
    end
    if (RDSMOD !=2 && RSourceGeo >0) begin
        Pdiss = Pdiss + V(s, si) * V(s, si) / Rsource;
    end
    Pwr(t) <+ delTemp1 * gth + ddt(delTemp1 * cth) - Pdiss;
end else begin
    Temp(t) <+ 0.0;
end
```

28. GDS Op-pt definition

Issue :

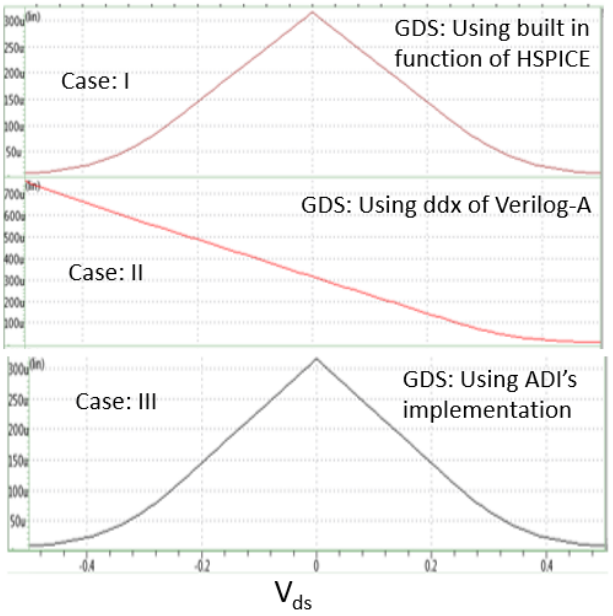
- GDS using ddx of Verilog A (Case II) is not symmetric. It should be symmetric as shown in case I.

BSIM-BULK106.2.0

```
GDS    = ddx(IDS, V(di));
```

BSIM-BULK107.0.0

```
if (sigvds > 0)begin
GDS    = ddx(IDS, V(di));
end
else begin
GDS    = ddx(-IDS, V(si));
end
```

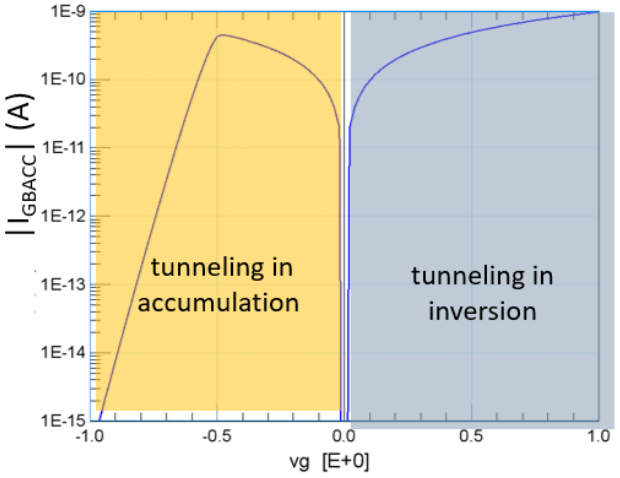


Gummel Symmetry Test circuit is used for biasing scheme in all cases.

29. Incorrect IGBBACC behavior in the accumulation region

Two Issues :

- Unphysical behavior in accumulation region
- High contribution of accumulation model in inversion region



- The value V_{auxacc} for calculating I_{gbacc} is calculated from

$$V_{auxacc} = NIGBACC * V_t * \log\left(1 + \exp\left(-\frac{V_{oxacc}}{NIGBACC * V_t}\right)\right)$$

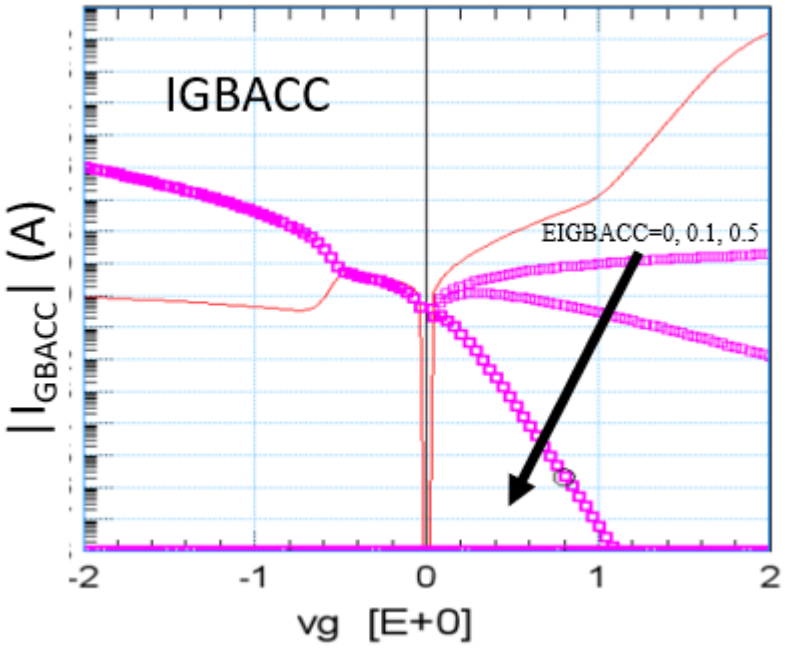
- For $V_{oxacc} = 0$, V_{auxacc} yields a constant value $NIGBACC * V_t * \log(2)$, but for any large enough positive V_{oxacc} the exp-function approaches 0 and thus the log-function 0 as well. Therefore, V_{auxacc} is constant > 0 for $V_{ox} > 0$ (towards inversion) but exponentially approaches 0 for $V_{ox} < 0$ (towards accumulation).
- This leads to the undesired behavior that V_{auxacc} has some finite value in inversion but 0 for accumulation. Therefore, wrong minus sign for V_{auxacc} should be removed

BSIM-BULK106.2.0

$$V_{auxacc} = NIGBACC * V_t * \log\left(1 + \exp\left(-\frac{V_{oxacc}}{NIGBACC * V_t}\right)\right)$$

BSIM-BULK107.0.0_Beta4

$$V_{auxacc} = NIGBACC * V_t * log\left(1 + exp\left(+\frac{V_{oxacc} - EIGBACC * 0.05\left(Vg + \sqrt{V_g^2 + 0.01}\right)}{NIGBACC * V_t}\right)\right)$$



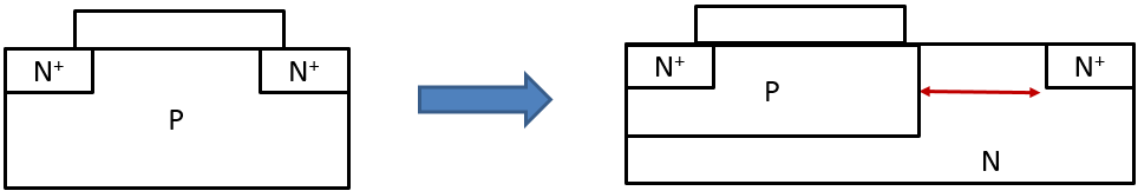
Note: IGBACC formulation is further updated in BSIM107.0.0

30. Enhancement for High Voltage modeling

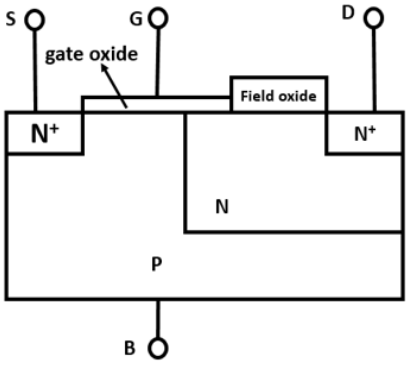
Wide application domain: Display, self-driving cars, etc.

To withstand high voltage:

- Increase gate oxide thickness
- Add a drift region between drain/gate: prevents breakdown of gate oxide and breakdown of drain junction.



Physics of Drift Region



Transport in the drift

$$I_{dr} = Q_{dr} * v_{dr}$$

$$I_{dr} = I_{ds}$$

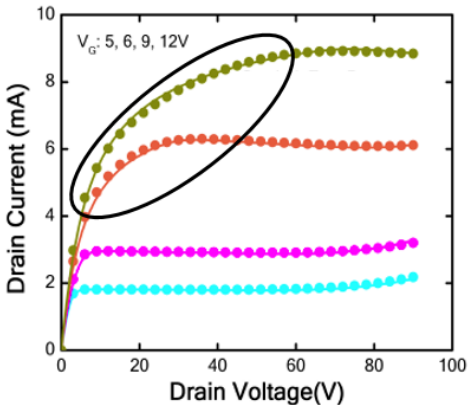
To support higher current, carrier velocity in the drift region increases

$$V_g \uparrow \rightarrow I_{ds} \uparrow \rightarrow v_{dr} \uparrow$$

As the carrier velocity reaches the saturation velocity limit, the **resistance of the drift region increases**

Physics of Drift Region

Ids-Vds of 90V transistor



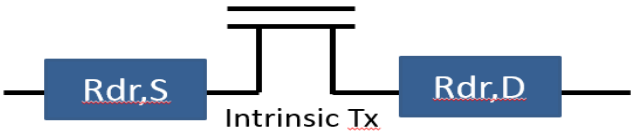
$$I_{dr} = W * Q_{dr} * v$$
$$= W * Q_{dr} * \mu * \frac{E}{1 + E/ESAT}$$

On integration,

$$R_{dr} = \frac{V_{dr}}{I_{dr}} = \frac{R0}{1 - R0 * \frac{I_{drift}}{ESAT * L_{drift}}} = \frac{R0}{1 - \frac{I_{dr}}{I_{dr,max}}}$$

$$R0 = \frac{L_{dr}}{W * \mu * Q_{drift}} \qquad I_{drift,max} = Q_{dr} * W * VSAT$$

Compact Model Adoption



$$R_{dr,D} = \frac{RDLCW}{\left[1 - \delta_{HV}\left\{\frac{I_{ds}}{I_{dr,sat,D}}\right\}^{MDRIFT}\right]^{\frac{1}{MDRIFT}}}$$

$$I_{dr,sat,D} = q * NDRIFTD * W_{eff} * VDRIFT$$

RDLCW: **R**esistance of the **D**rain side at **L**ow **C**urrent

MDRIFT: Smoothing parameter for velocity saturation

VDRIFT: Saturation Velocity in the drift

NDRIFTD: Charge Density in the drift

δ_{HV} introduced for smoothness. Nominal value ~ 1

Source side parameters: RSLCW, NDRIFTS

Implementation in BSIM-BULK

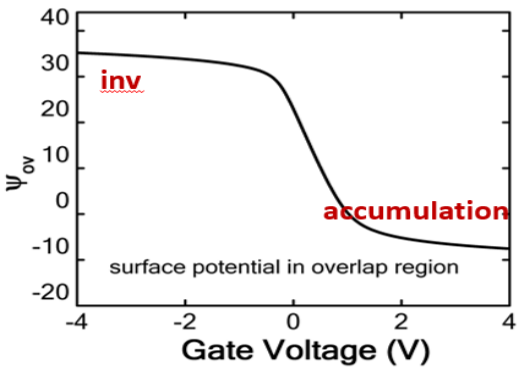
- Turn-key feature: Activates only when *switch HVMOD is set to 1.*
- Default value of HVMOD is 0 (HV feature turned-off)

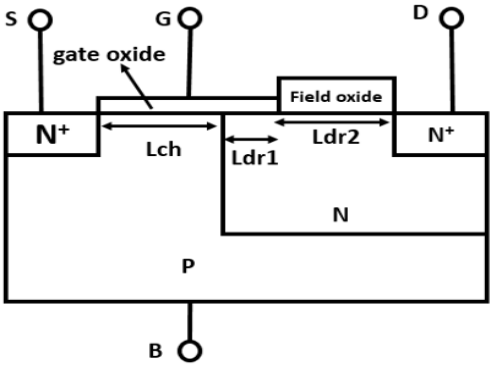
```
if (RDSMOD== 1 && HVMOD == 1) begin
    T4  = 1 + PDRWB * Vbsx;
    T0  = ids ;
    T11 = NF * Weff * `q  * VDRIFT_t ;
    if (RDLCW!=0) begin
        idrift sat d = T11 * NDRIFTD  ;
```

```
rdrift d = rdstemphv * RDLCW * WeffWRFactor/T2D * T4;

Rdrain  = Rdrain + rdrift_d;
Rsource = Rsource + rdrift_s;
```

Charge Model





$$V_{gdi} - V_{fb} = \psi_P + \gamma * \sqrt{e_P^{-\psi} + \psi_P - 1}$$ **(Solved analytically)**

$$Q_{I,dr} = W * L_{dr1} * 2n_q * C_{ox} * V_t * q_{dr}$$

$$Q_{B,dr} = W * L_{dr1} * 2n_q * C_{ox} * V_t * (V_{gdi} - V_{fb} - \psi_{s,ov} - 2n_q * q_{dr})$$

Model Implementation

- Activate the model: *Set HVCAP=1 along with HVMOD =1*
- Default value of HVCAP is 0

```
if (HVCAP == 1 && HVMOD == 1) begin
   // CV calculations

   vgfbdrift = -devsign * V(g,di) - VFBOV ;
   vgfbdrift = vgfbdrift/Vt;

   gamhv     = sqrt(2.0 * `q * epssi * NDR * inv_Vt) / Cox;
   phibHV    = lln(NDR / ni);

   `PO_psip(vgfbdrift,gamhv,0,phibHV,psip_k)
   `BSIM_q(psip_k, phibHV, devsign *V(di,b)/Vt, gamhv, q_k)

  QBOV = NF * Wact * LOVER * `EPS0 * EPSROX / BSIMBULKTOXP * Vt *
  QIOV = NF * Wact * LOVERACC * 2 * nq_hv * Vt * T0 * q_k ;

      Qovb = Qovb + QIOV;
      Qovd = Qovd + QBOV;
```

Parameters introduced for the charge model:

VFBOV: Flat-band voltage of the drift region
LOVER: Length of the drift region
LOVERACC: Effective length in accumulation
NDR: Doping of the drift region
SLHV: Parameter and Flag for smoothing the capacitance
SLHV1: Parameter for smoothing the capacitance

If HVCAPS==1, overlap capacitance model is also added to the source side

31. Warning Message for U0 ≤ 0

In present VA code implementation, there is no warning/error message to report negative U0.

BSIM-BULK106.2.0

```
`MPRnb( U0 ,67.0e-3    ,"m^2/V/s"    ,"Low Field mobility." )
```

BSIM-BULK107.0.0

```
`MPRoz( U0 ,67.0e-3    ,"m^2/V/s"    ,"Low Field mobility." )
```

BSIM-BULK107.0.0

```
if (U0_i <= 0.0) begin
    $strobe("Warning: U0_i = %e is non-positive, setting it to the default value.", U0_i);
    U0_i = 0.067;
end
if (UA_i < 0.0) begin
    $strobe("Warning: UA_i = %e is negative, setting it to 0.", UA_i);
    UA_i = 0.0;
end
if (EU_i < 0.0) begin
    $strobe("Warning: EU_i = %e is negative, setting it to 0.", EU_i);
    EU_i = 0.0;
end
if (UD_i < 0.0) begin
    $strobe("Warning: UD_i = %e is negative, setting it to 0.", UD_i);
    UD_i = 0.0;
end
if (UCS_i < 0.0) begin
    $strobe("Warning: UCS_i = %e is negative, setting it to 0.", UCS_i);
    UCS_i = 0.0;
end
```

32. Range of Model Parameters

BSIM-BULK107.0.0_Beta4

```
`MPRcz( SLHV1     ,1.0    ,""    ,"Parameter for slope of the accumulation capacitance " )

    if (SLHV > 0) begin
        T1 = 1 + q_k / SLHV1 ;
        T2 = SLHV * 1.9e-9 / T1;
        T0 = 3.9 * `EPS0 / (BSIMBULKTOXP * 3.9 / EPSROX + T2 / epsratio);
    end else begin
        T0 = `EPS0 * EPSROX / BSIMBULKTOXP;
    end
```

There is a possible division by 0 when SLHV1=0.
It will be restricted to >0

BSIM-BULK107.0.0

```
`MPRoz( SLHV1     ,1.0    ,""    ,"Parameter for slope of the accumulation capacitance " )
```

BSIM-BULK107.0.0_Beta4

```
`MPRnb( NDRIFTD        ,5.0e16          ,"per m^2"          ,"Charge density in the drift region" )

                Nextra = Ntot/NDRIFTD - 1;
```

MPRnb means "no bound“ but charge density should not be negative. It will be restricted to >0.

BSIM-BULK107.0.0

```
`MPRoz( NDRIFTD    ,5.0e16      ,"per m^2"     ,"Charge density in the drift region" )
```

33. Default value of NDEPEDGE

BSIM-BULK106.2.0

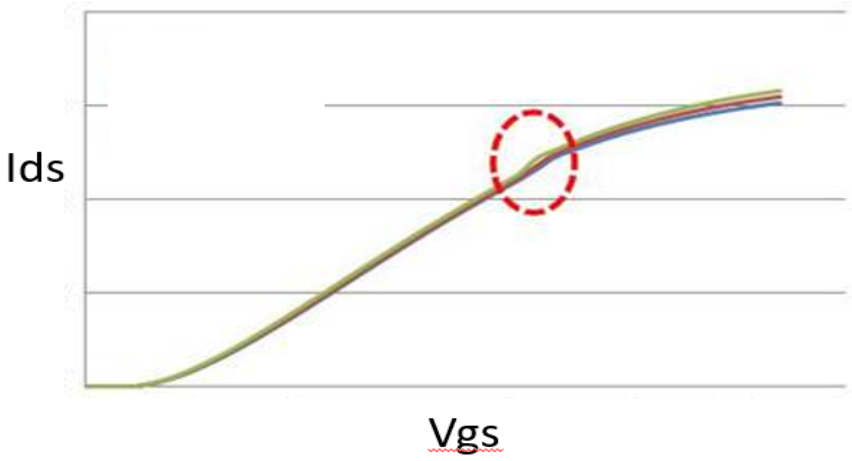
```
`MPRnb( NDEPEDGE          ,NDEP    ,"1/m^3"      ,"Channel doping concentration for EDGEFET" )
```

BSIM-BULK107.0.0

```
`MPRnb( NDEPEDGE    ,1e24       ,"1/m^3"     ,"Channel doping concentration for EDGEFET" )
```

34. Kink in DC Current due to PSATB

During evaluation of Beta4, There is a kink in current.

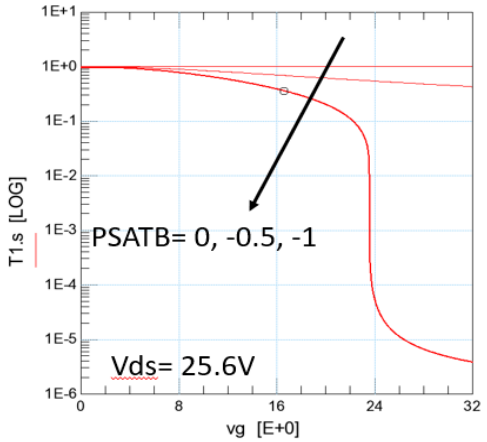


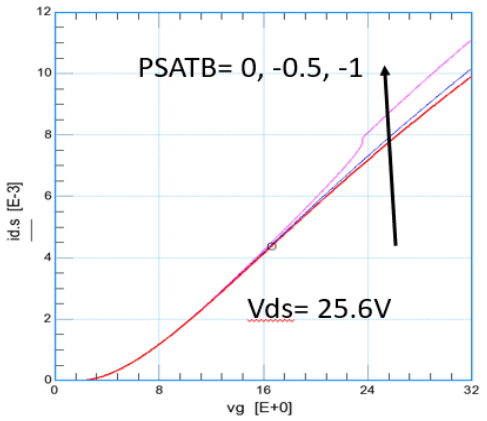
The kink in current is due to PSTAB implementation

BSIM-BULK107.0.0_Beta1

$$x = PSATB * VBSX \qquad y = 1 - x$$

$$T1 = 0.5\left(y + \sqrt{y^2 + (4 * 10^{-6})}\right)$$





T1 is having a sharp dip from 1 to 1e-6 at Vg ≈ 24V for PSATB = -1.

BSIM-BULK107.0.0_Beta1

```
T1 = hypsmooth((1.0 - PSATB_i * Vbsx), 1.0e-3);
```

BSIM-BULK107.0.0

```
T11 = PSATB_i * Vbsx;
T12 = sqrt(0.1+T11*T11);
T1  = 0.5*(1-T11+sqrt((1-T11)*(1-T11)+T12));
```

BSIM-BULK107.0.0

$$x = PSATB * VBSX$$

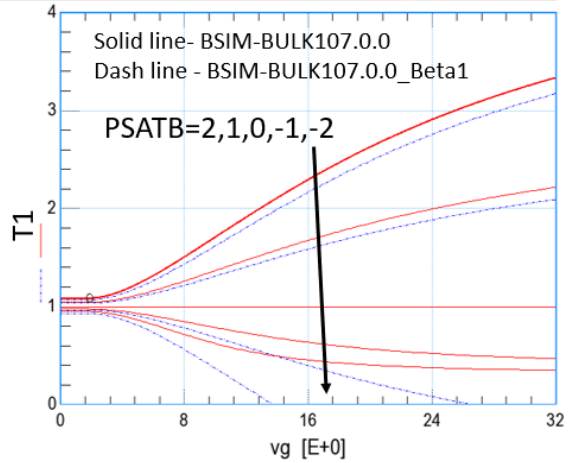
$$y = 1 - x$$

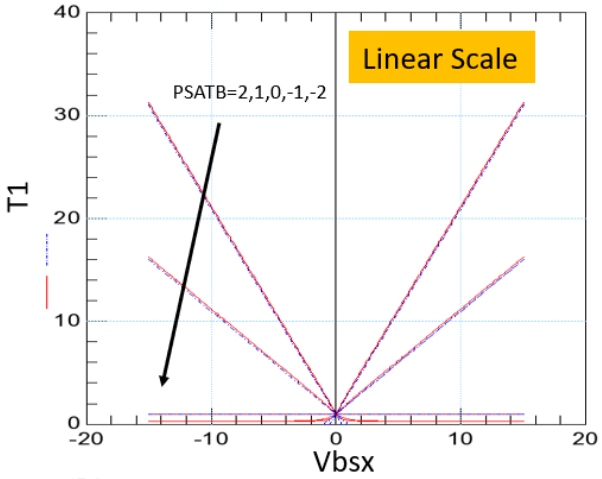
BSIM-BULK107.0.0_Beta1

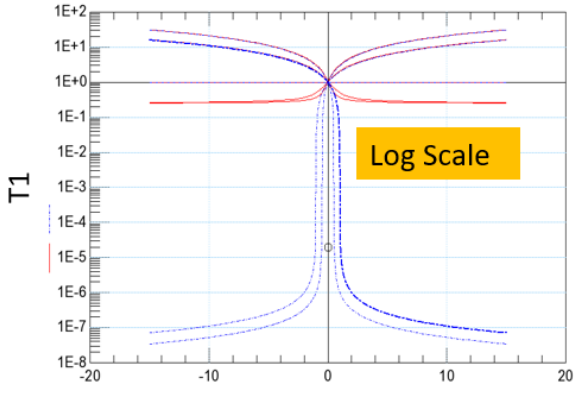
$$T1 = 0.5\left(y + \sqrt{y^2 + (4 * 10^{-6})}\right)$$

BSIM-BULK107.0.0

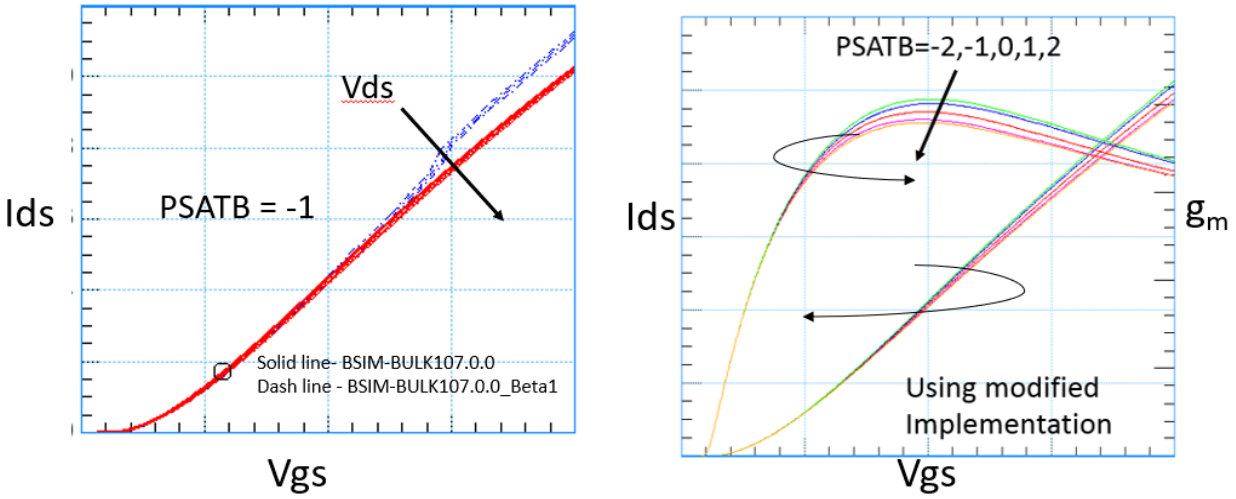
$$T1 = 0.5\left(y + \sqrt{y^2 + \sqrt{0.1 + x * x}}\right)$$







Drain current with improved implementation of PSATB:



35. Rollback MNUD and MNUD1 from thermal noise model

2019enh1: Impact of C0 on thermal noise (TSMC)

Issue:

- C0 significantly affect thermal noise behavior.
- TSMC wants there should be no effect of non zero C0 on thermal noise.

Proposed Solution:

- MNUD and MNUD1 are also added in the thermal noise Sid such that non zero C0 does not affect thermal noise; see next slide for simulation results.

```
// MNUD model to enhance Id-Vd fitting flexibility
T0   = (qs + qdeff);
T1   = (qs - qdeff);
T2   = T1 / (T0 + M0_t);
T3   = K0_t * T2 * T2;
Mnud = 1.0 + T3;

// MNUD1 to enha
T9    = C0_t / (max(0, C0SAT_t - T1 - T1) - T0 + 2.0 - T - Vth);
Mnud1 = lexp(-T9);
Dtot  = Dmob * Dvsat * Dr;
```

TSMC suggested to remove this update as this has not been tested.

- Original Implementation:

  Ids=k1*(qs+qdeff)*MNUD*MNUD1;

  Sid=K2*(qs+qdeff);

- Proposed Implementation:

  This update was made in 107.0.0-beta3

  Ids=k1*(qs+qdeff)*MNUD*MNUD1;

  Sid=K2*(qs+qdeff)*MNUD*MNUD1;

p. 1

In thermal noise module:

BSIM-BULK107.0.0_Beta3

```
T0    = ueff * Dptwg * Moc * Cox * Vtn;
T1    = 0.5 * (qs + qdeff)* Mnud1 * Mnud;
T3    = T1 + 0.5;
T4    = T3 * T3;
T5    = T4 * T3;
T6    = (qs - qdeff)* Mnud1 * Mnud;
```

BSIM-BULK107.0.0

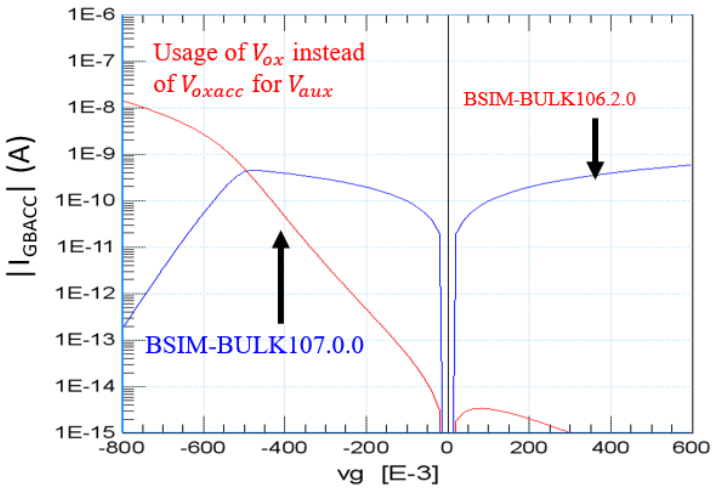
```
T0    = ueff * Dptwg * Moc * Cox * Vtn;
T1    = 0.5 * (qs + qdeff);
T3    = T1 + 0.5;
T4    = T3 * T3;
T5    = T4 * T3;
T6    = (qs - qdeff);
```

36. Addressing IGBACC issue

$$V_{auxacc} = NIGBACC * V_t * \log\left(1 + \exp\left(-\frac{V_{oxacc}}{NIGBACC * V_t}\right)\right)$$ BSIM-BULK106.2.0

- Instead of using V_{oxacc} in calculation of V_{auxacc}, use V_{OX} with corrected sign similar to BSIM-CMG.

$$V_{auxacc} = NIGBACC * V_t * \log\left(1 + \exp\left(-\frac{V_{ox}}{NIGBACC * V_t}\right)\right)$$ BSIM-BULK107.0.0



When IGBMOD=1 :

BSIM-BULK106.2.0

```
T1      = Voxmacc / NIGBACC_i / Vt;
Vaux_Igbacc = NIGBACC_i * Vt * lln(1.0 + lexp(-T1));
```

BSIM-BULK107.0.0_Beta4

```
T7      = sqrt(nVt * vg *nVt * vg + 1.0e-4);
T8      = 0.5*EIGBACC_i*(nVt * vg+T7);
T1      = (Voxmacc-T8) / NIGBACC_i / Vt;
Vaux_Igbacc = NIGBACC_i * Vt * lln(1.0 + lexp(T1));
```

BSIM-BULK107.0.0

```
T1      = Voxm / NIGBACC_i / Vt;
Vaux_Igbacc = NIGBACC_i * Vt * lln(1.0 + lexp(-T1));
```

37.Add GMIN across D-B and S-B junctions

- "gmin" is supposed to be a simulator-controlled parameter, it must not be a model parameter.

BSIM-BULK107.0.0

```
real gmin;
gmin = $simparam("gmin", 1e-12);
```

BSIM-BULK106.2.0

```
if (RBODYMOD != 0) begin
    I(sbulk, si) <+ devsign * Ibs;
    I(dbulk, di) <+ devsign * Ibd;
    I(sbulk, si) <+ devsign * ddt(Qbsj);
    I(dbulk, di) <+ devsign * ddt(Qbdj);
end else begin
    I(bi, si) <+ devsign * Ibs;
    I(bi, di) <+ devsign * Ibd;
    I(bi, si) <+ devsign * ddt(Qbsj);
    I(bi, di) <+ devsign * ddt(Qbdj);
end
```

BSIM-BULK107.0.0

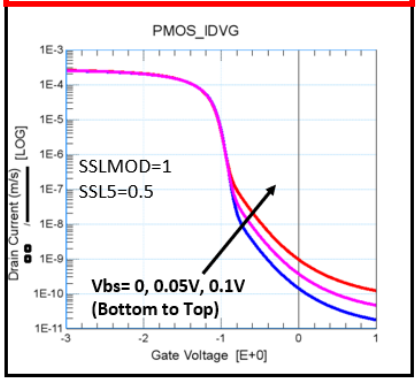
```
if (RBODYMOD != 0) begin
    I(sbulk, si) <+ devsign * Ibs + V(sbulk, si) * gmin;
    I(dbulk, di) <+ devsign * Ibd + V(dbulk, di) * gmin;
    I(sbulk, si) <+ devsign * ddt(Qbsj);
    I(dbulk, di) <+ devsign * ddt(Qbdj);
end else begin
    I(bi, si) <+ devsign * Ibs + V(bi, si) * gmin;
    I(bi, di) <+ devsign * Ibd  + V(bi, di) * gmin;
    I(bi, si) <+ devsign * ddt(Qbsj);
    I(bi, di) <+ devsign * ddt(Qbdj);
end
```

38. Missing Type in Sub-Surface Leakage Model

```
if (SSLMOD != 0) begin
    T1 = pow(NDEP_i / 1.0e23, SSLEXP1);
    T2 = pow(300.0 / DevTemp, SSLEXP2);
    T3 = (SSL5 * V(b,s)) / Vt;
    SSL0_NT  = SSL0 * lexp(-T1 * T2);
    SSL1_NT  = SSL1 * T2 * T1;
    PHIB_SSL = SSL3 * tanh(lexp(devsign * SSL4 * (V(gi, bi) - VTH - V(s,b))));
    Issl     = sigvds * NF * Weff * SSL0_NT * lexp(T3) * lexp(-SSL1_NT * Leff)
               * lexp(PHIB_SSL / Vt) * (lexp(SSL2 * Vdsx / Vt) - 1.0);
    I(di, si) <+ devsign * Issl;
end
```

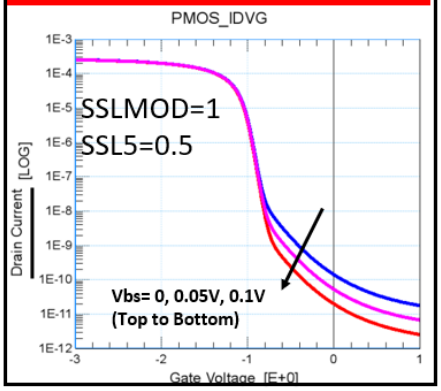
- Sub-Surface leakage current should decrease with increase in reverse VB (positive for PMOS).
- To capture the correct trend with VB, there should be "devsign" in equation T3

```
T3 = (SSL5 * V(b,s)) / Vt;
```



BSIM-BULK107.0.0

```
T3 = (devsign *SSL5 * V(b,s)) / Vt;
```



39. External/Internal Node Issue in Sub-Surface Leakage Model

BSIM-BULK107.0.0_Beta2

```
if (SSLMOD != 0) begin
    T1 = pow(NDEP_i / 1.0e23, SSLEXP1);
    T2 = pow(300.0 / DevTemp, SSLEXP2);
    T3 = (SSL5 * V(b,s)) / Vt;
    SSL0_NT  = SSL0 * lexp(-T1 * T2);
    SSL1_NT  = SSL1 * T2 * T1;
    PHIB_SSL = SSL3 * tanh(lexp(devsign * SSL4 * (V(gi, bi) - VTH - V(s,b))));
    Issl     = sigvds * NF * Weff * SSL0_NT * lexp(T3) * lexp(-SSL1_NT * Leff) * lexp(PHIB_SSL / Vt) * (lexp(SSL2 * Vdsx / Vt) - 1.0);
    I(di, si) <+ devsign * Issl;
end
```

BSIM-BULK107.0.0

```
if (SSLMOD != 0) begin
    T1 = pow(NDEP_i / 1.0e23, SSLEXP1);
    T2 = pow(300.0 / DevTemp, SSLEXP2);
    T3 = (devsign*SSL5 * V(bi,si)) / Vt;
    SSL0_NT  = SSL0 * lexp(-T1 * T2);
    SSL1_NT  = SSL1 * T2 * T1;
    PHIB_SSL = SSL3 * tanh(lexp(devsign * SSL4 * (V(gi, bi) - VTH - V(si,bi))));
    Issl     = sigvds * NF * Weff * SSL0_NT * lexp(T3) * lexp(-SSL1_NT * Leff) * lexp(PHIB_SSL / Vt) * (lexp(SSL2 * Vdsx / Vt) - 1.0);
    I(di, si) <+ devsign * Issl;
end
```

40. HVMOD Issue- CV Model

- In V(g,di) or V(g,si), it should be 'gm' instead of 'g' (g means external gate node).
- Regarding V(di,b) and V(si,b), It is expected that it should be bi (body network is independent of HVMOD)

- QBOV:

BSIM-BULK107.0.0_Beta5

`BSIM_q(psip_k, phibHV, devsign *V(di,b)/Vt, gamhv, q_k)
vgfbdrift = -devsign * V(g,di) - VFBOV

BSIM-BULK107.0.0

`BSIM_q(psip_k, phibHV, devsign *V(di,bi)/Vt, gamhv, q_k)
vgfbdrift = -devsign * V(gm,di) - VFBOV

- QBOVS: BSIM-BULK107.0.0_Beta5

`BSIM_q(psip_k, phibHV, devsign *V(si,b)/Vt, gamhv, q_k)
vgfbdrift = -devsign * V(g,si) - VFBOV

BSIM-BULK107.0.0

`BSIM_q(psip_k, phibHV, devsign *V(si,bi)/Vt, gamhv, q_k)
vgfbdrift = -devsign * V(gm,si) - VFBOV

41. Secondary impact ionization in HVMOD

— In secondary impact ionization in the drift region, Bias V(d, b), d, b means external drain and body node, it should be bi (body network is independent of HVMOD).

`Smooth(devsign * V(d,b) - Vdseff - DRII2, 0, 0.05, T2) BSIM-BULK107.0.0_Beta5

`Smooth(devsign * V(d,bi) - Vdseff - DRII2, 0, 0.05, T2) BSIM-BULK107.0.0

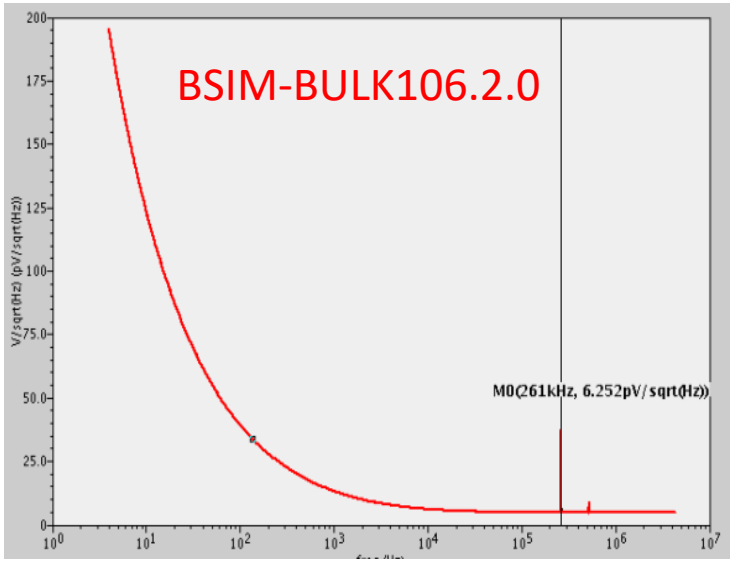
42. Implementing ‘sigvds’ in flicker noise

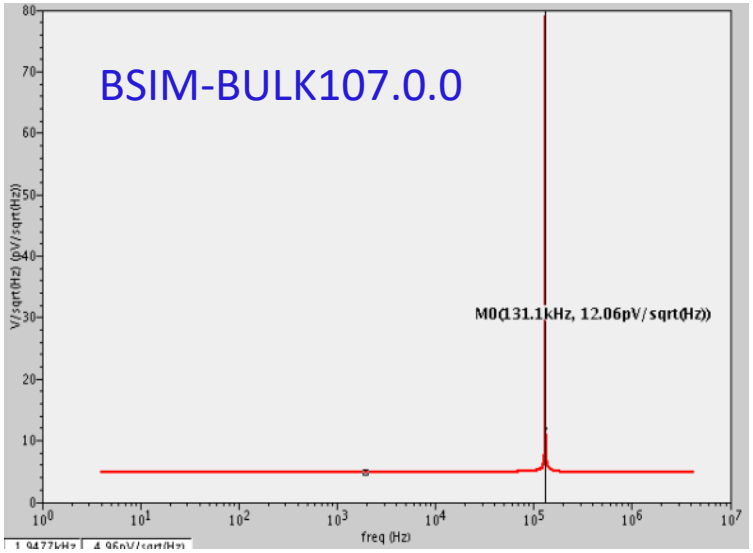
BSIM-BULK106.2.0

```
I(di, si) <+ flicker_noise(FNPowerAt1Hz, EF, "loverf");
```

BSIM-BULK107.0.0

```
I(di, si) <+ flicker_noise(sigvds*FNPowerAt1Hz, EF, "loverf");
```





For more detail please refer [1]

[1] G. J. Coram, et. al., “Flicker Noise Formulations in Compact Models,” Transaction on Computer-Aided Design of integrated Circuits and Systems, 13 January 2020.