Summary of Changes in BSIM-BULK106.2.0:

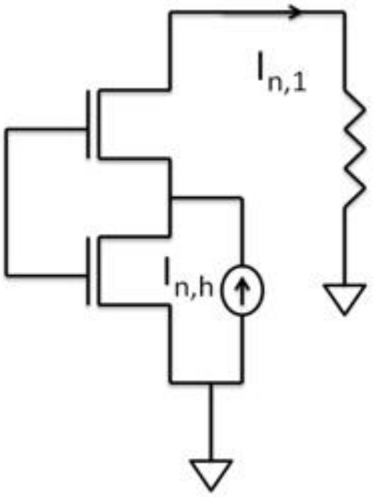
- Removed all the "ifdef" statements from the code.
- Node collapsing has been done for the nodes “t” and “N1” for "SHMOD = 0" and "TNOIMOD = 0".
- Every "IF" and "else" statement starts with "begin" and ends with the "end" statement.
- Modified the name of the parameter "MULUO" to "MULU0".
- Name of the Variable "T13" has modified to Weff_ SH in the code.
- Modified Flicker noise model which can also model the flicker noise trends for Halo transistors.
- MNUD model to increase flexibility in I_{DS} – V_{DS} fitting.
- Improved Gamma behavior in the linear Region for TNOIMOD = 1.
- Modified Thermal noise model to tune NF50 in Subthreshold region and to achieve ideal trend of gamma for long channel transistors.
- Clamping added in I_{GD} and I_{GS}.
- Sign correction in charge calculations.
- Added Self-Heating output variables.
- Deactivated node “T” when RTH0 or SHMOD or both are zero.
- Used “SRCFLAG” as the formal argument of **BSIM6RdsEndSha** and **BSIM6RdseffGeo** in place of “TYPE”.

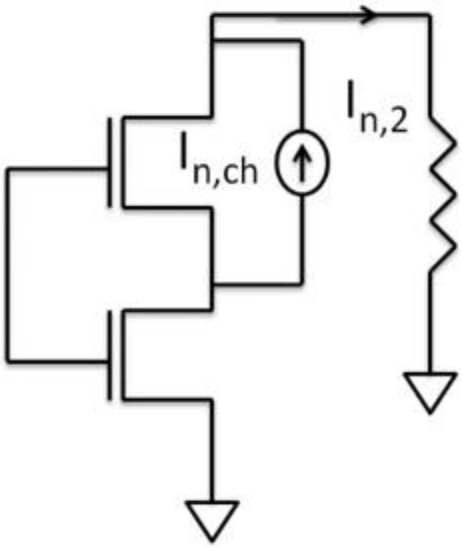
- Modified Calculations of Geometry-dependent source/drain resistance.
- Added **devsign** in Operating point VTH expression.
- Limit (Weff + WTH0) to (Weff + WTH0) > 0 in Self Heating module.
- Changes made in the macros definition.
- Module name has changed from **"bsim6"** to **"bsimbulk".**
- Absence of CLM and velocity saturation terms in Charge expressions in manual.
- Parameter DVTP0 has Re-Defined in Manual.
- Added binning parameters in the EDGEFET parameters.
- Default value of EDGEFET parameter has changed to "0" from "1".
- Modified the expression of built in potential.
- DTEMP, MULU0, DELVTO and IDS0MULT have modified to both instance and model parameters.
- Removed all the "ifdef" statements from the code.
- Added units and descriptions for all the parameters.
- Added modified function of V_{DSX} to avoid negative G_{DS} issue.
- Removed clamping from UCR.
- Added protection to the following parameters: LP1, LP2, NJS, NJD, XJBV S and XJBV D.

1. Modified Flicker noise model which can also model the flicker noise trends for Halo transistors.

SOLUTION:

This model can be activated by making FNOINOD = 0





$$I_{n1} \cong I_{n,h} \frac{g_{m,ch} + g_{d,ch}}{g_{m,ch} + g_{d,ch} + g_{d,h}} \qquad I_{n2} \cong I_{n,ch} \frac{g_{d,h}}{g_{m,ch} + g_{d,ch} + g_{d,h}}$$

$$S_{ID,1} \cong S_{ID,h} \underbrace{\left[\frac{g_{m,ch} + g_{d,ch}}{g_{m,ch} + g_{d,ch} + g_{d,h}}\right]^2}_{CF_h} \qquad S_{ID,2} \cong S_{ID,ch} \underbrace{\left[\frac{g_{d,h}}{g_{m,ch} + g_{d,ch} + g_{d,h}}\right]^2}_{CF_{ch}}$$

$$\boldsymbol{S_{ID} = S_{ID,1} + S_{ID,2}}$$

$$\boxed{\boldsymbol{S_{ID} = S_{ID,h}.CF_{\mathrm{h}} + S_{ID,ch}.CF_{ch}}}$$

$$S_{ID,h} = \frac{KTI_{DS}^2}{\gamma f W L_h^2} \int_0^{L_h} \frac{N_{t,h}^*(E_{F_n})}{N_h^2} dx$$

$$N_t^* = NOIA + NOIB.N + NOIC.N^2$$

$$S_{ID,ch} = \frac{KTI_{DS}^2}{\gamma f W (L - L_h)^2} \int_{L_h}^{L} \frac{N_{t,ch}^*(E_{F_n})}{N_{ch}^2} dx$$

$$i_h = \frac{I_{DS}}{2n_q \mu C_{ox} \frac{W}{L_h} V_t^2} = (q_{s,h}^2 + q_{s,h}) - (q_{d,h}^2 + q_{d,h})$$

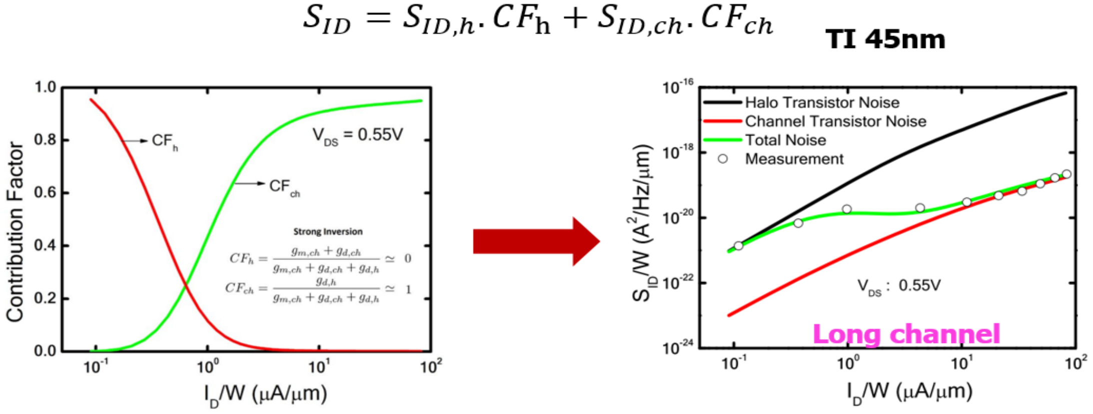
$$q_{d,h} = -\frac{1}{2} + \frac{1}{2}\sqrt{1 + 4(q_{s,h}^2 + q_{s,h} - i_h)}$$

similarly

$$i_{ch} = \frac{I_{DS}}{2n_q \mu C_{ox} \frac{W}{L - L_h} V_t^2} = (q_{s,ch}^2 + q_{s,ch}) - (q_{d,ch}^2 + q_{d,ch})$$

$$q_{s,ch} = -\frac{1}{2} + \frac{1}{2}\sqrt{1 + 4(q_{d,ch}^2 + q_{d,ch} + i_{ch})}$$

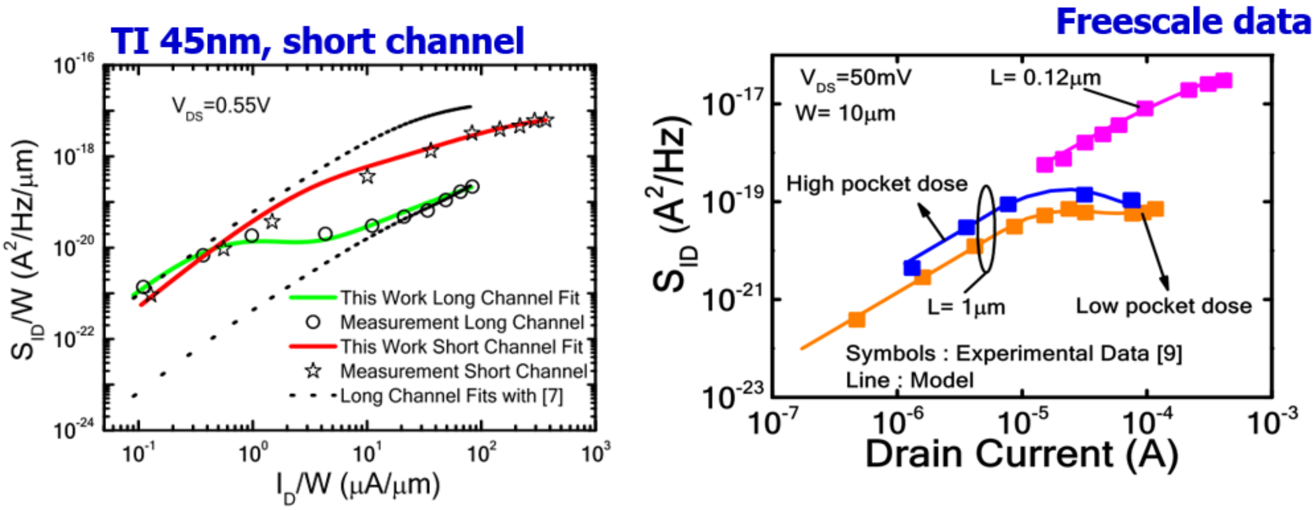
$$S_{ID} = S_{ID,h}.CF_{\text{h}} + S_{ID,ch}.CF_{ch}$$



Weak Inversion: $CF_h >> CF_{ch}$ --------> $S_{ID} = S_{ID,h}$

Strong Inversion: $CF_{ch} >> CF_h$ --------> $S_{ID} = S_{ID,ch}$

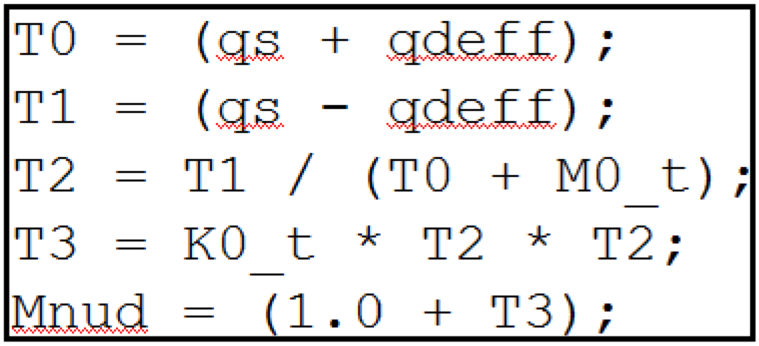
Characteristic bias dependency



H. Agarwal, Y. S Chauhan et al., “**Analytical modeling of Flicker Noise in Halo Implanted MOSFETs”, IEEE JEDS, Vol. 3, Issue 4, April 2015.**

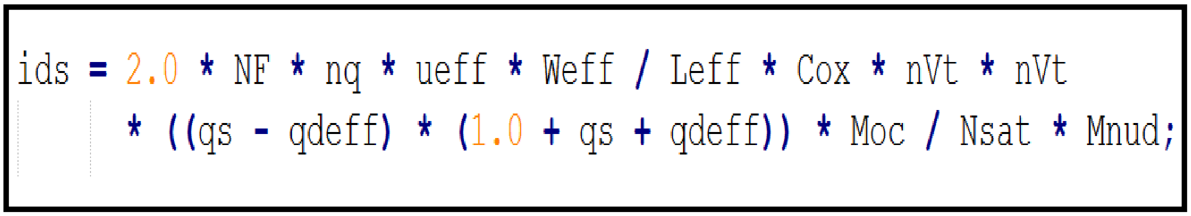
2. MNUD model to increase flexibility in I_{DS} – V_{DS} fitting.

SOLUTION:

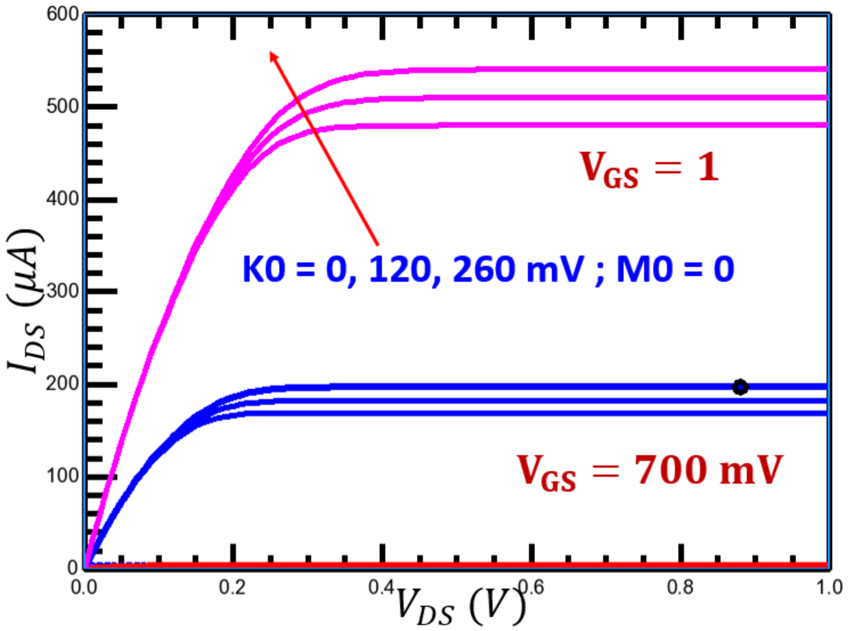


```
T0 = (qs + qdeff);
T1 = (qs - qdeff);
T2 = T1 / (T0 + M0_t);
T3 = K0_t * T2 * T2;
Mnud = (1.0 + T3);
```

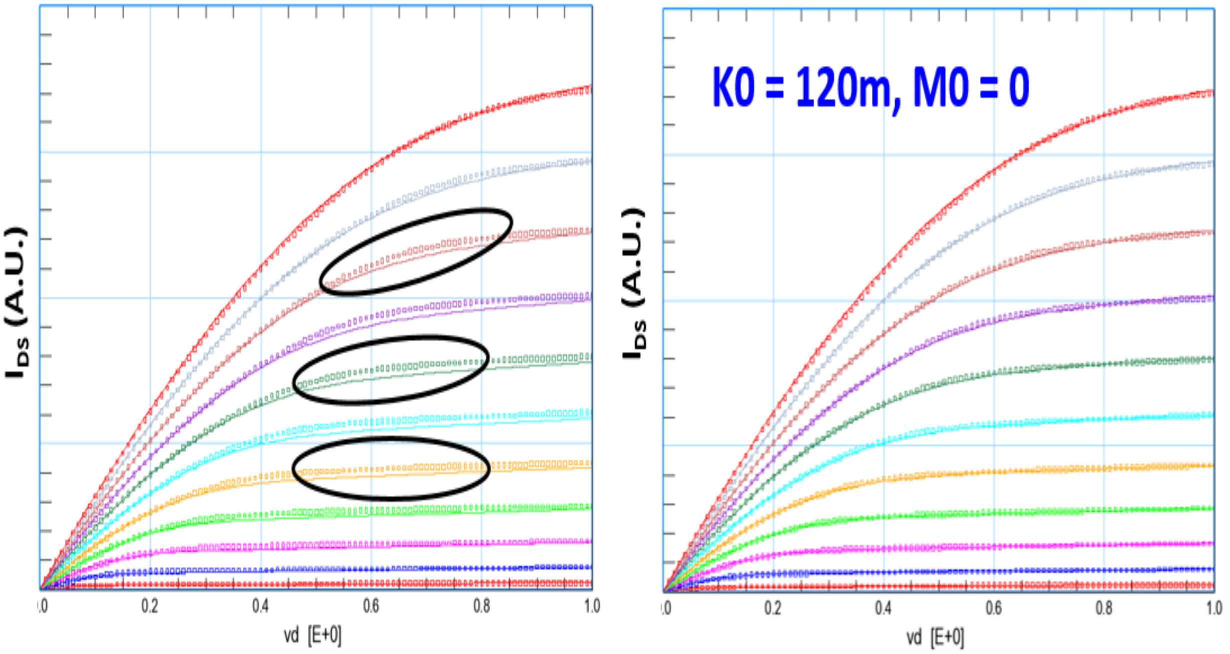
$$\mathbf{MNUD = 1 + K_0 * \left(\frac{q_s - q_{deff}}{M0 + q_S + q_{deff}}\right)^2}$$



```
ids = 2.0 * NF * nq * ueff * Weff / Leff * Cox * nVt * nVt
     * ((qs - qdeff) * (1.0 + qs + qdeff)) * Moc / Nsat * Mnud;
```



The effect of negative values of K0 is shown in the

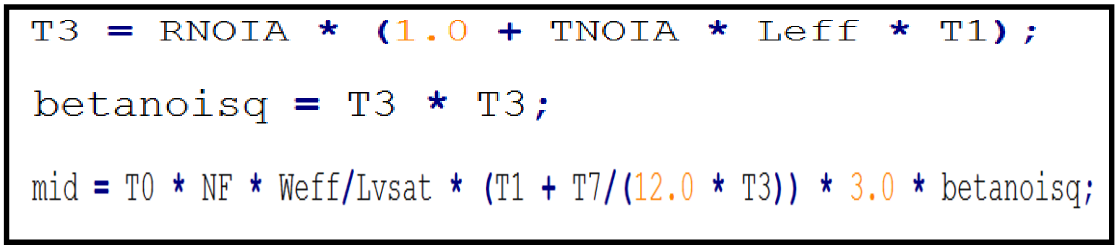


MNUD model improves the I_{DS}-V_{DS} fitting.

3. Improved gamma Behavior in Linear Region for TNOIMOD = 1.

SOLUTION:

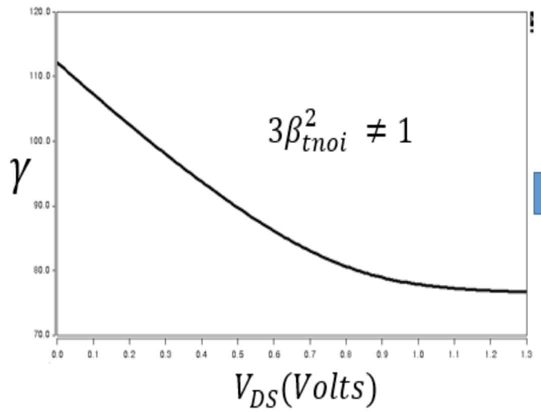
In BSIM6.1.1



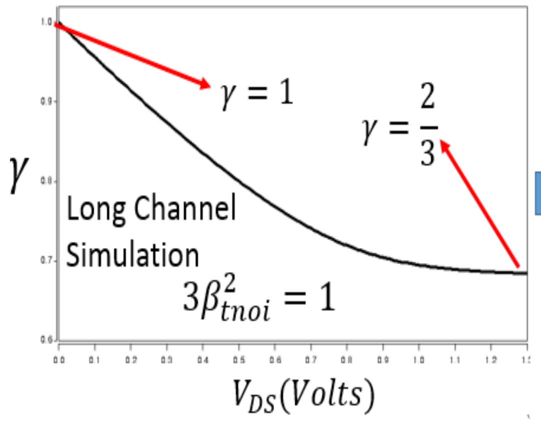
```
T3 = RNOIA * (1.0 + TNOIA * Leff * T1);

betanoisq = T3 * T3;

mid = T0 * NF * Weff/Lvsat * (T1 + T7/(12.0 * T3)) * 3.0 * betanoisq;
```

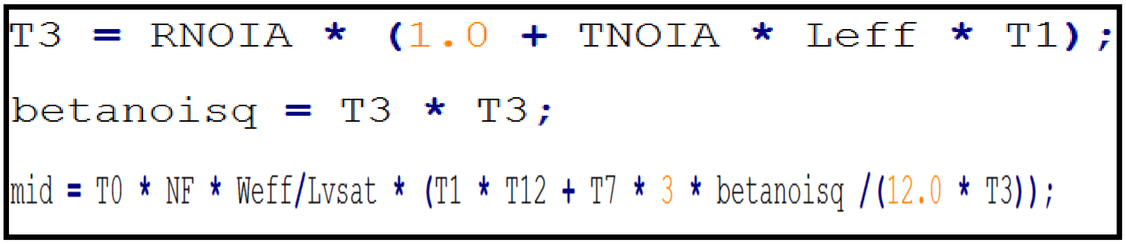


When $3\beta_{tnoi}^2 \neq 1$, parameters of $\boldsymbol{\beta}_{\mathbf{tnoi}}$ also affect the Linear Region, so couldn't follow the R_{EFF} Simulation.

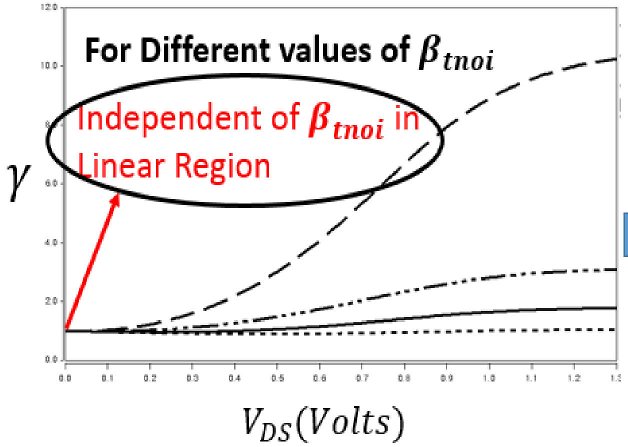


Following R_{EFF} Simulation when $3\beta_{tnoi}^2 = 1$

In BSIM-BULK106.2.0



```
T3 = RNOIA * (1.0 + TNOIA * Leff * T1);
betanoisq = T3 * T3;
mid = T0 * NF * Weff/Lvsat * (T1 * T12 + T7 * 3 * betanoisq /(12.0 * T3));
```

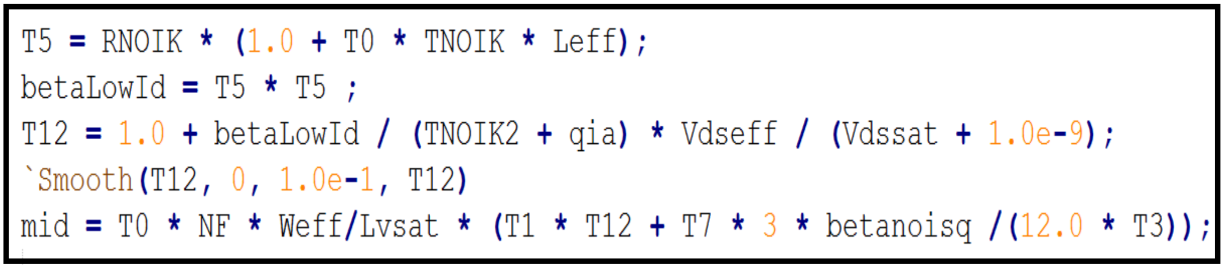


γ vs V_{DS} plots for different values of β_{tnoi}. γ is independent of β_{tnoi} in the linear region and hence follows the R_{EFF} simulation in Linear region.

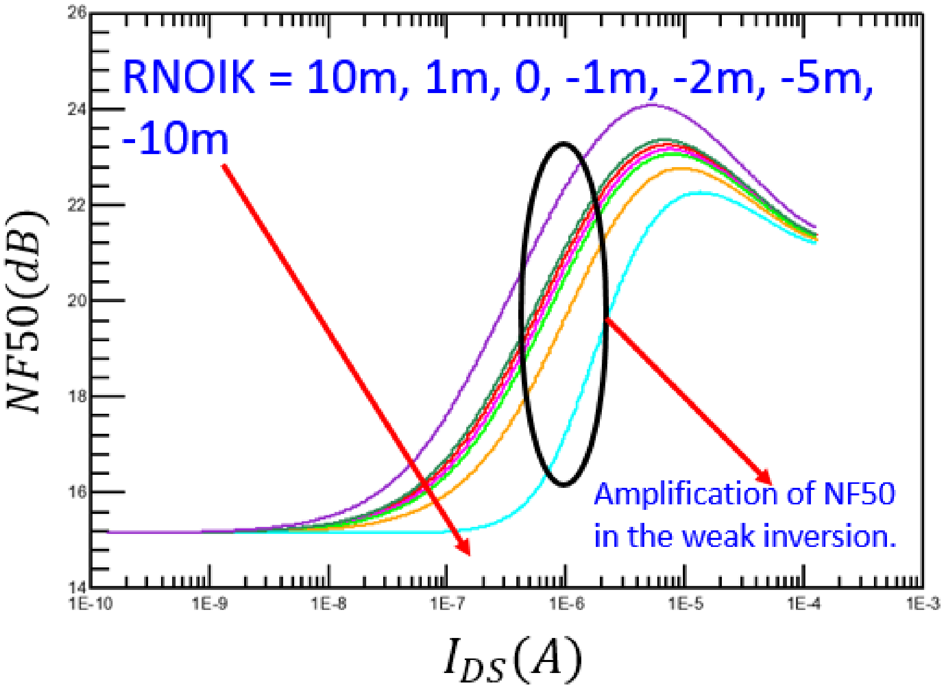
4. Modified Thermal Noise model to tune NF50 in Subthreshold region and to achieve ideal trend of gamma for long channel transistors.

SOLUTION:

In BSIM-BULK106.2.0



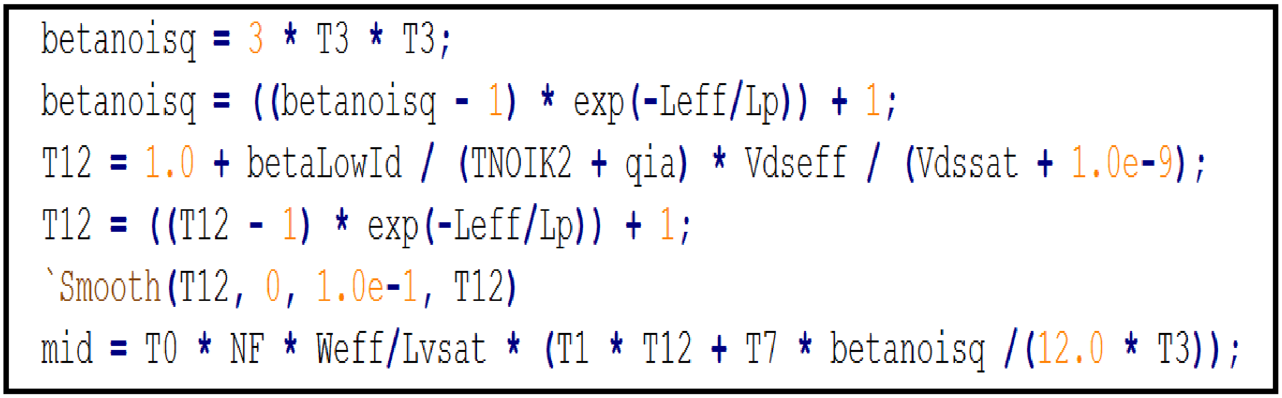
```
T5 = RNOIK * (1.0 + T0 * TNOIK * Leff);
betaLowId = T5 * T5 ;
T12 = 1.0 + betaLowId / (TNOIK2 + qia) * Vdseff / (Vdssat + 1.0e-9);
`Smooth(T12, 0, 1.0e-1, T12)
mid = T0 * NF * Weff/Lvsat * (T1 * T12 + T7 * 3 * betanoisq /(12.0 * T3));
```



RNOIK = 10m, 1m, 0, -1m, -2m, -5m, -10m

Amplification of NF50 in the weak inversion.

Ideal trend of gamma for long channel transistors.



```
betanoisq = 3 * T3 * T3;
betanoisq = ((betanoisq - 1) * exp(-Leff/Lp)) + 1;
T12 = 1.0 + betaLowId / (TNOIK2 + qia) * Vdseff / (Vdssat + 1.0e-9);
T12 = ((T12 - 1) * exp(-Leff/Lp)) + 1;
`Smooth(T12, 0, 1.0e-1, T12)
mid = T0 * NF * Weff/Lvsat * (T1 * T12 + T7 * betanoisq /(12.0 * T3));
```

where L_P is a model parameter

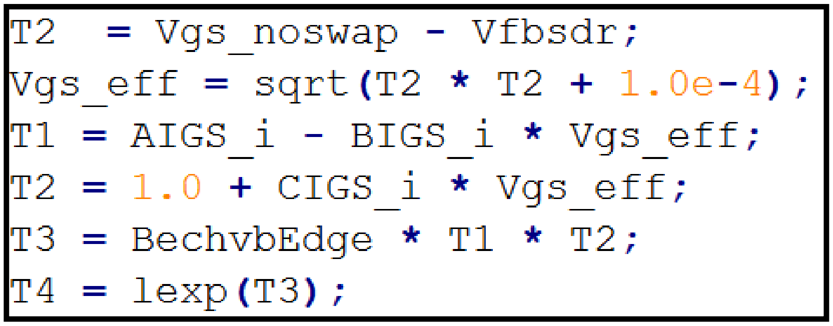
$$L < L_P \; (normal\,region) \Longrightarrow 3\beta_{tnoi}^{2}(or\; T12)$$

$$L > L_P \Longrightarrow 1$$

5. Clamping of I_{GD} and I_{GS}.

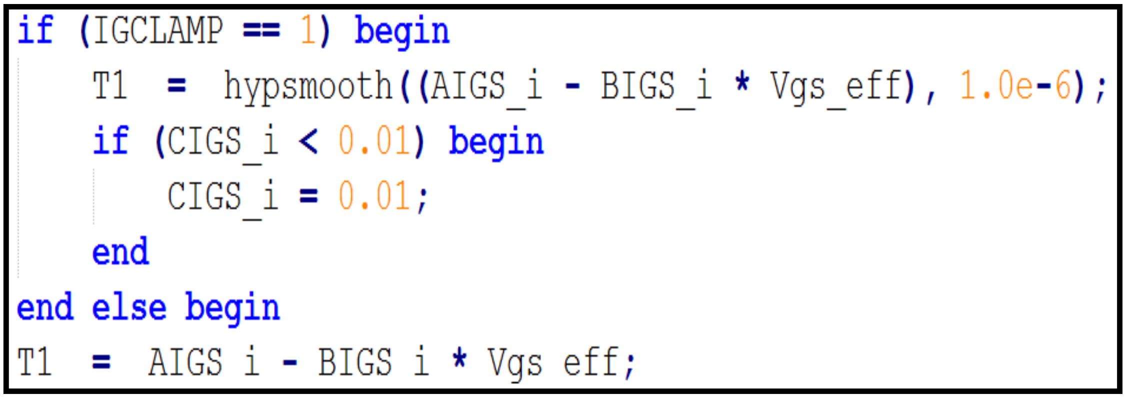
SOLUTION:

In BSIM6.1.1



```
T2   = Vgs_noswap - Vfbsdr;
Vgs_eff = sqrt(T2 * T2 + 1.0e-4);
T1 = AIGS_i - BIGS_i * Vgs_eff;
T2 = 1.0 + CIGS_i * Vgs_eff;
T3 = BechvbEdge * T1 * T2;
T4 = lexp(T3);
```

In BSIM-BULK106.2.0



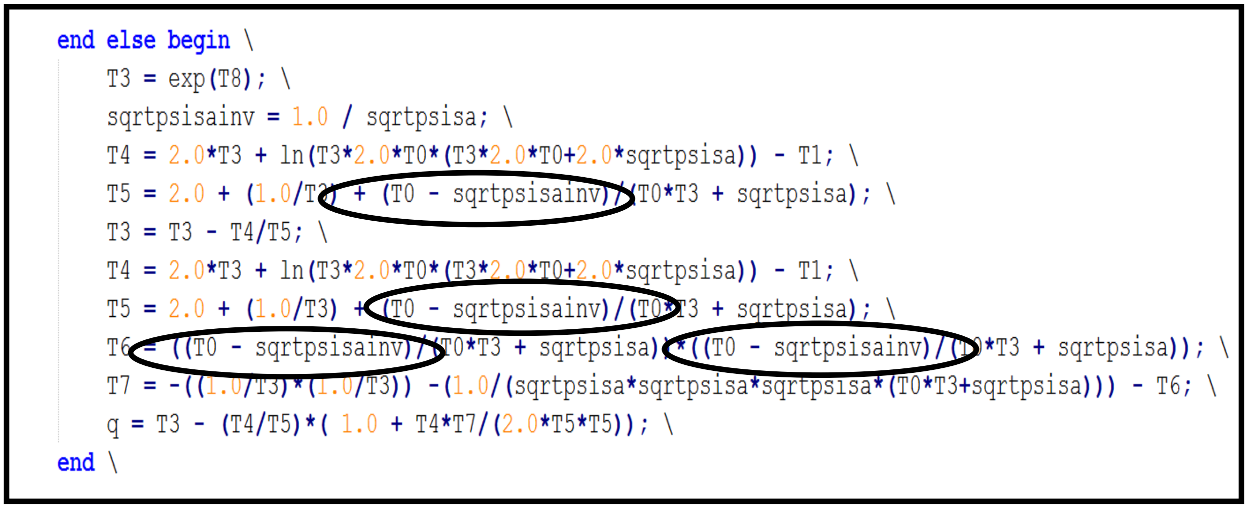
```
if (IGCLAMP == 1) begin
    T1  =  hypsmooth((AIGS_i - BIGS_i * Vgs_eff), 1.0e-6);
    if (CIGS_i < 0.01) begin
        CIGS_i = 0.01;
    end
end else begin
T1  =  AIGS i - BIGS i * Vgs eff;
```

Similarly, clamping for I_{gd} has also added.

6. Sign Correction in charge Calculations.

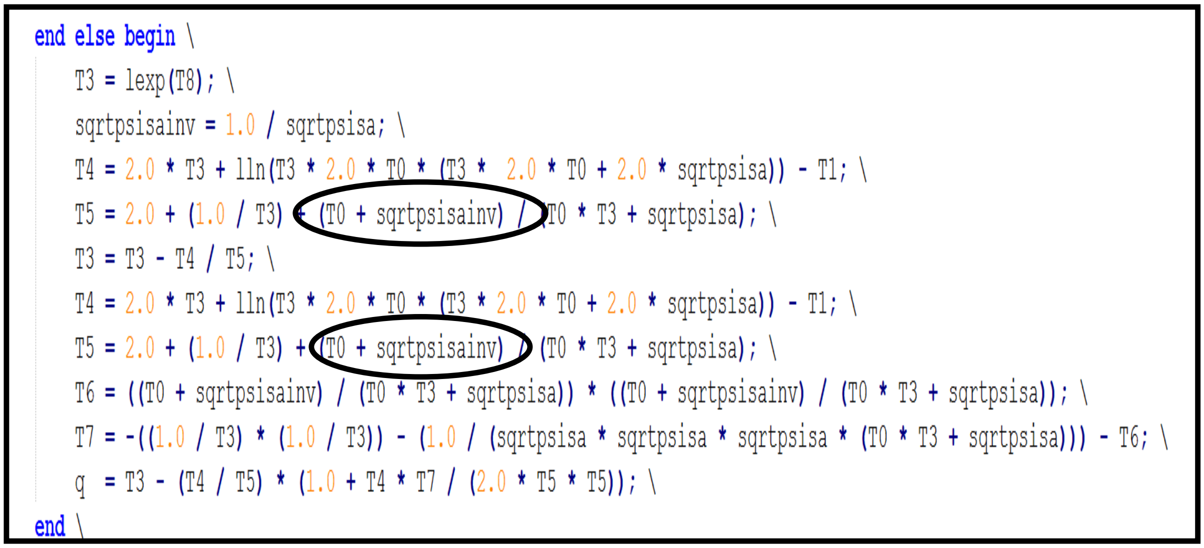
SOLUTION:

In BSIM6.1.1



```
end else begin \
    T3 = exp(T8); \
    sqrtpsisainv = 1.0 / sqrtpsisa; \
    T4 = 2.0*T3 + ln(T3*2.0*T0*(T3*2.0*T0+2.0*sqrtpsisa)) - T1; \
    T5 = 2.0 + (1.0/T3) + (T0 - sqrtpsisainv)/(T0*T3 + sqrtpsisa); \
    T3 = T3 - T4/T5; \
    T4 = 2.0*T3 + ln(T3*2.0*T0*(T3*2.0*T0+2.0*sqrtpsisa)) - T1; \
    T5 = 2.0 + (1.0/T3) + (T0 - sqrtpsisainv)/(T0*T3 + sqrtpsisa); \
    T6 = ((T0 - sqrtpsisainv)/(T0*T3 + sqrtpsisa))*((T0 - sqrtpsisainv)/(T0*T3 + sqrtpsisa)); \
    T7 = -((1.0/T3)*(1.0/T3)) -(1.0/(sqrtpsisa*sqrtpsisa*sqrtpsisa*(T0*T3+sqrtpsisa))) - T6; \
    q = T3 - (T4/T5)*( 1.0 + T4*T7/(2.0*T5*T5)); \
end \
```

In BSIM-BULK106.2.0

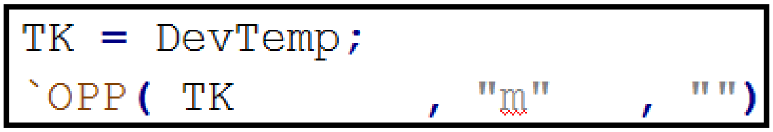


```
end else begin \
    T3 = lexp(T8); \
    sqrtpsisainv = 1.0 / sqrtpsisa; \
    T4 = 2.0 * T3 + lln(T3 * 2.0 * T0 * (T3 *  2.0 * T0 + 2.0 * sqrtpsisa)) - T1; \
    T5 = 2.0 + (1.0 / T3) + (T0 + sqrtpsisainv) / (T0 * T3 + sqrtpsisa); \
    T3 = T3 - T4 / T5; \
    T4 = 2.0 * T3 + lln(T3 * 2.0 * T0 * (T3 * 2.0 * T0 + 2.0 * sqrtpsisa)) - T1; \
    T5 = 2.0 + (1.0 / T3) + (T0 + sqrtpsisainv) / (T0 * T3 + sqrtpsisa); \
    T6 = ((T0 + sqrtpsisainv) / (T0 * T3 + sqrtpsisa)) * ((T0 + sqrtpsisainv) / (T0 * T3 + sqrtpsisa)); \
    T7 = -((1.0 / T3) * (1.0 / T3)) - (1.0 / (sqrtpsisa * sqrtpsisa * sqrtpsisa * (T0 * T3 + sqrtpsisa))) - T6; \
    q  = T3 - (T4 / T5) * (1.0 + T4 * T7 / (2.0 * T5 * T5)); \
end \
```

7. Added Self-Heating output variables.

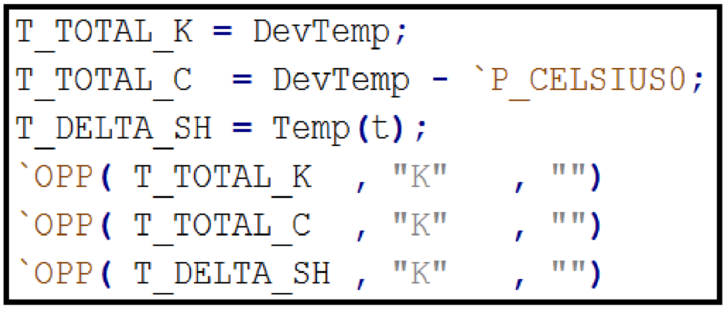
SOLUTION:

In BSIM6.1.1



```
TK = DevTemp;
`OPP( TK          , "m"      , "")
```

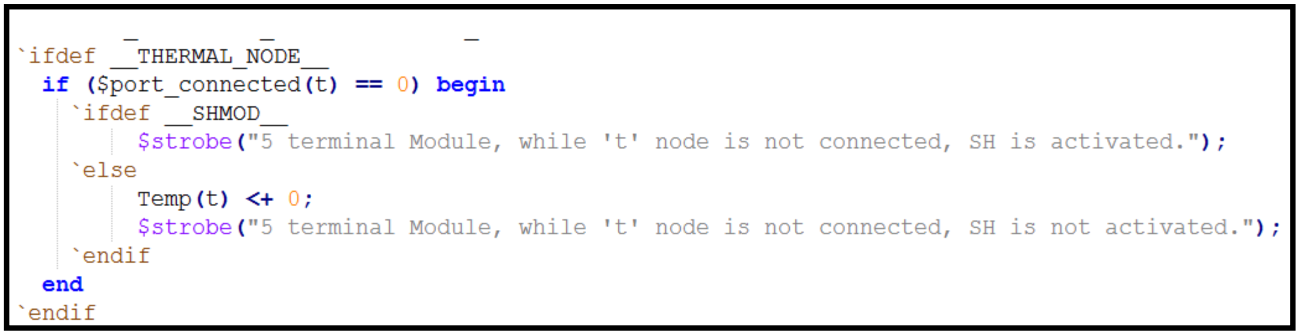
In BSIM-BULK106.2.0



```
T_TOTAL_K = DevTemp;
T_TOTAL_C  = DevTemp - `P_CELSIUS0;
T_DELTA_SH = Temp(t);
`OPP( T_TOTAL_K  , "K"   , "")
`OPP( T_TOTAL_C  , "K"   , "")
`OPP( T_DELTA_SH , "K"   , "")
```

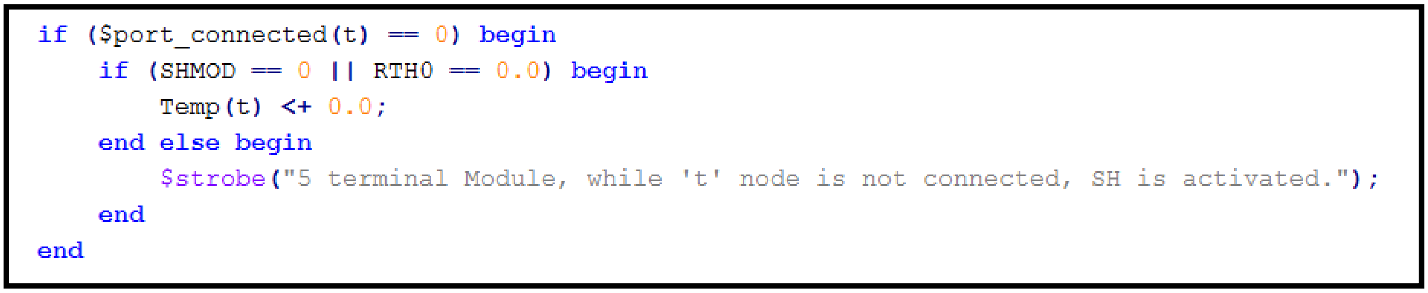
8. Deactivated node T when RTH0 or SHMOD or both are zero.

In BSIM6.1.1



```
`ifdef __THERMAL_NODE__
  if ($port_connected(t) == 0) begin
    `ifdef __SHMOD__
        $strobe("5 terminal Module, while 't' node is not connected, SH is activated.");
    `else
        Temp(t) <+ 0;
        $strobe("5 terminal Module, while 't' node is not connected, SH is not activated.");
    `endif
  end
`endif
```

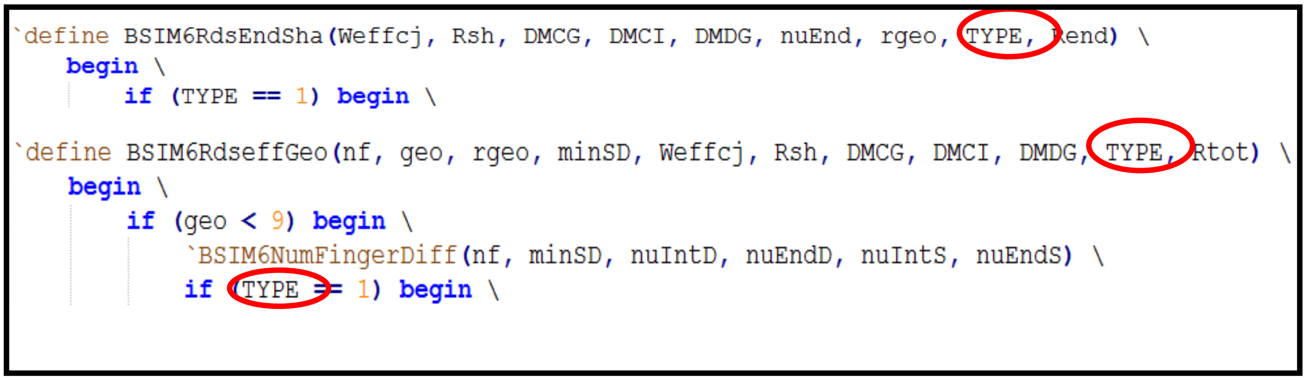
In BSIM-BULK106.2.0



```
if ($port_connected(t) == 0) begin
    if (SHMOD == 0 || RTH0 == 0.0) begin
        Temp(t) <+ 0.0;
    end else begin
        $strobe("5 terminal Module, while 't' node is not connected, SH is activated.");
    end
end
```

9. Used “SRCFLAG” in place of “TYPE” as the formal argument of BSIM6RdsEndSha and BSIM6RdseffGeo

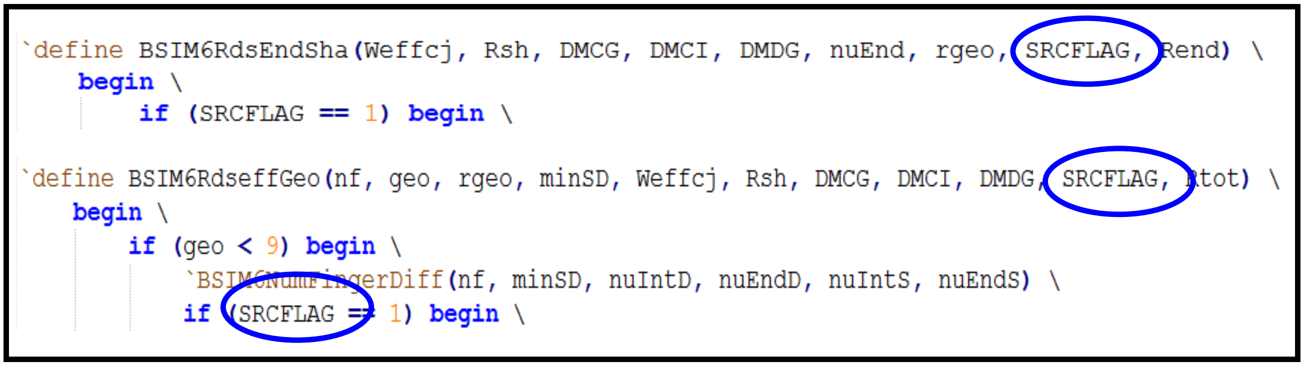
In BSIM6.1.1



```
`define BSIM6RdsEndSha(Weffcj, Rsh, DMCG, DMCI, DMDG, nuEnd, rgeo, TYPE, Rend) \
    begin \
        if (TYPE == 1) begin \

`define BSIM6RdseffGeo(nf, geo, rgeo, minSD, Weffcj, Rsh, DMCG, DMCI, DMDG, TYPE, Rtot) \
    begin \
        if (geo < 9) begin \
            `BSIM6NumFingerDiff(nf, minSD, nuIntD, nuEndD, nuIntS, nuEndS) \
            if (TYPE == 1) begin \
```

In BSIM-BULK106.2.0

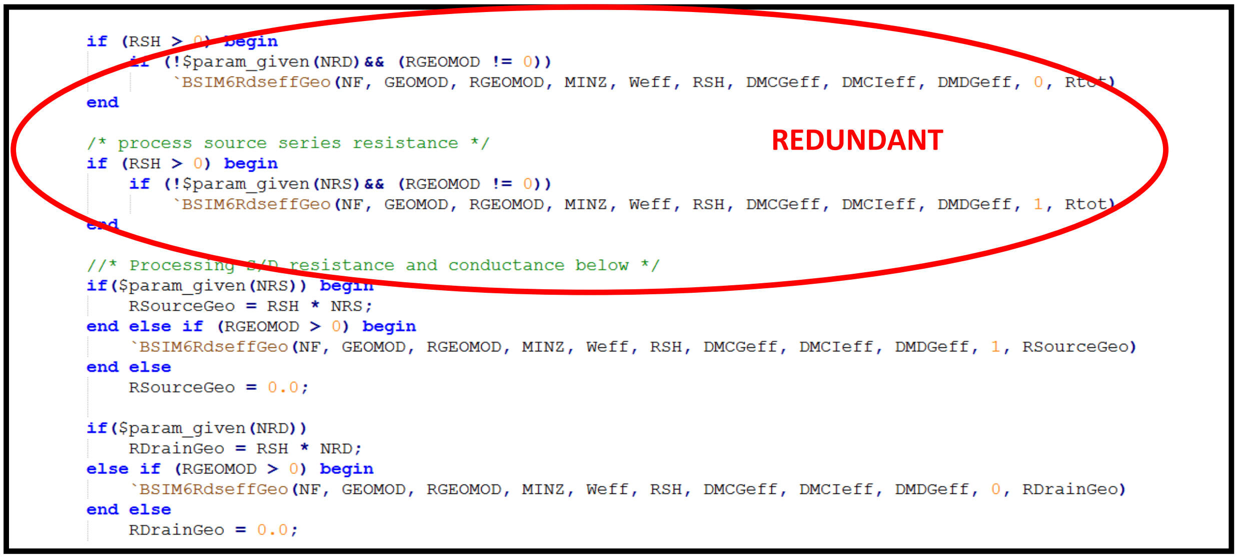


```
`define BSIM6RdsEndSha(Weffcj, Rsh, DMCG, DMCI, DMDG, nuEnd, rgeo, SRCFLAG, Rend) \
    begin \
        if (SRCFLAG == 1) begin \

`define BSIM6RdseffGeo(nf, geo, rgeo, minSD, Weffcj, Rsh, DMCG, DMCI, DMDG, SRCFLAG, Rtot) \
    begin \
        if (geo < 9) begin \
            `BSIM6NumFingerDiff(nf, minSD, nuIntD, nuEndD, nuIntS, nuEndS) \
            if (SRCFLAG == 1) begin \
```

10. Modified Calculations of Geometry-dependent source/drain resistance.

In BSIM6.1.1



REDUNDANT

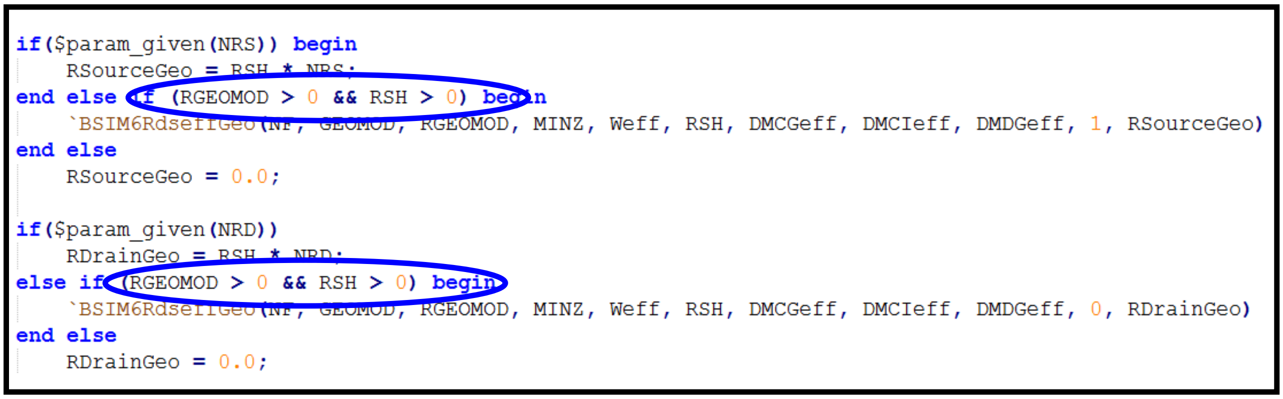
```
if (RSH > 0) begin
    if (!$param_given(NRD)&& (RGEOMOD != 0))
        `BSIM6RdseffGeo(NF, GEOMOD, RGEOMOD, MINZ, Weff, RSH, DMCGeff, DMCIeff, DMDGeff, 0, Rtot)
end

/* process source series resistance */
if (RSH > 0) begin
    if (!$param_given(NRS)&& (RGEOMOD != 0))
        `BSIM6RdseffGeo(NF, GEOMOD, RGEOMOD, MINZ, Weff, RSH, DMCGeff, DMCIeff, DMDGeff, 1, Rtot)
end

//* Processing S/D resistance and conductance below */
if($param_given(NRS)) begin
    RSourceGeo = RSH * NRS;
end else if (RGEOMOD > 0) begin
    `BSIM6RdseffGeo(NF, GEOMOD, RGEOMOD, MINZ, Weff, RSH, DMCGeff, DMCIeff, DMDGeff, 1, RSourceGeo)
end else
    RSourceGeo = 0.0;

if($param_given(NRD))
    RDrainGeo = RSH * NRD;
else if (RGEOMOD > 0) begin
    `BSIM6RdseffGeo(NF, GEOMOD, RGEOMOD, MINZ, Weff, RSH, DMCGeff, DMCIeff, DMDGeff, 0, RDrainGeo)
end else
    RDrainGeo = 0.0;
```

In BSIM-BULK106.2.0

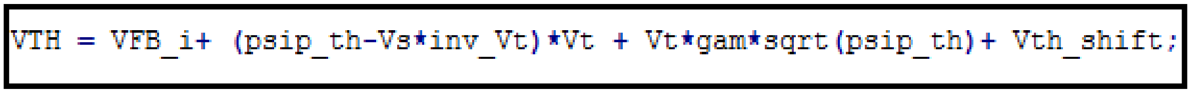


```
if($param_given(NRS)) begin
    RSourceGeo = RSH * NRS;
end else if (RGEOMOD > 0 && RSH > 0) begin
    `BSIM6RdseffGeo(NF, GEOMOD, RGEOMOD, MINZ, Weff, RSH, DMCGeff, DMCIeff, DMDGeff, 1, RSourceGeo)
end else
    RSourceGeo = 0.0;

if($param_given(NRD))
    RDrainGeo = RSH * NRD;
else if (RGEOMOD > 0 && RSH > 0) begin
    `BSIM6RdseffGeo(NF, GEOMOD, RGEOMOD, MINZ, Weff, RSH, DMCGeff, DMCIeff, DMDGeff, 0, RDrainGeo)
end else
    RDrainGeo = 0.0;
```

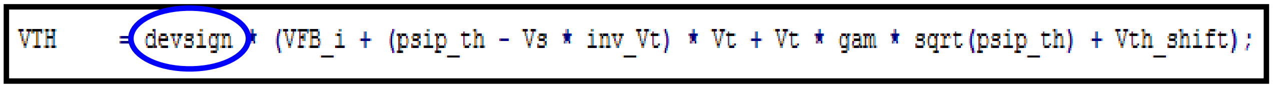
11. Added devsign in Operating point V_{TH} expression.

In BSIM6.1.1

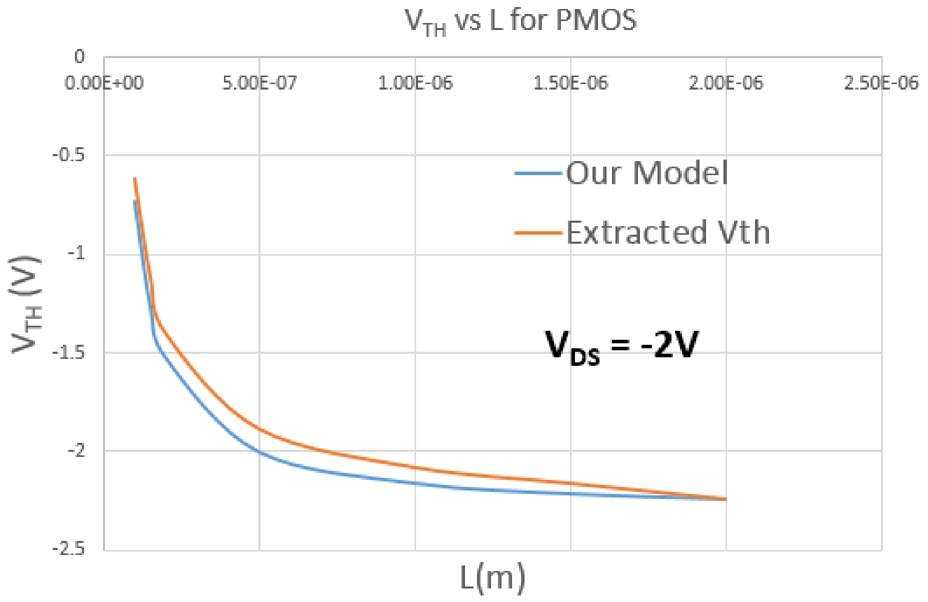


```
VTH = VFB_i+ (psip_th-Vs*inv_Vt)*Vt + Vt*gam*sqrt(psip_th)+ Vth_shift;
```

In BSIM-BULK106.2.0

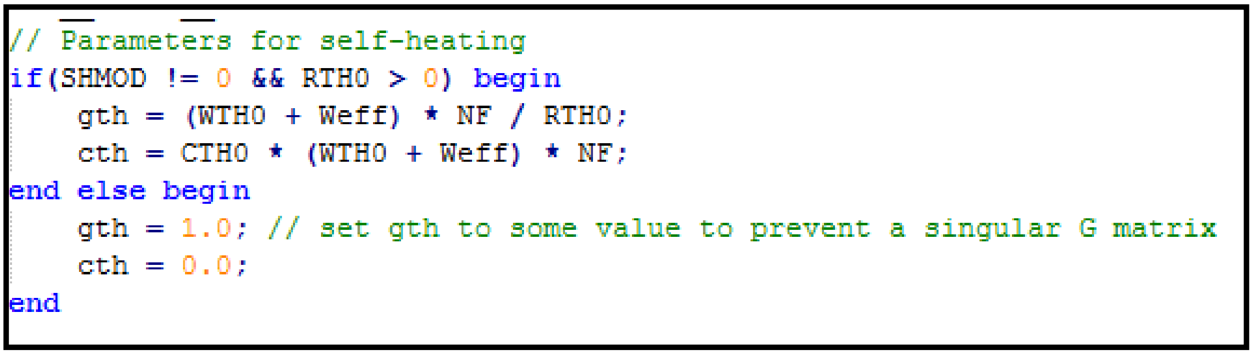


```
VTH      = devsign * (VFB_i + (psip_th - Vs * inv_Vt) * Vt + Vt * gam * sqrt(psip_th) + Vth_shift);
```



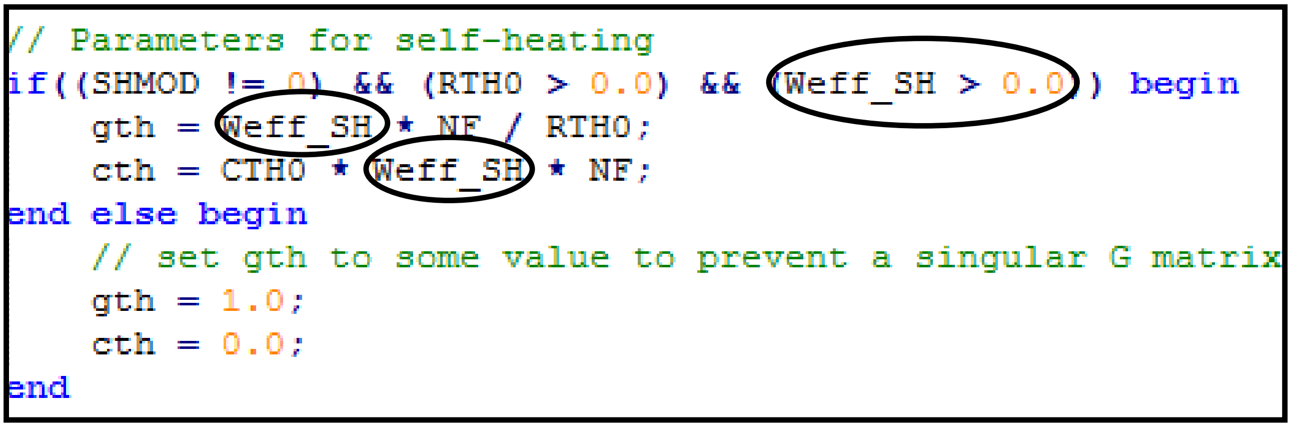
12. Limit (Weff + WTH0) to (Weff + WTH0) > 0 in Self Heating module.

In BSIM6.1.1



```
// Parameters for self-heating
if(SHMOD != 0 && RTH0 > 0) begin
    gth = (WTH0 + Weff) * NF / RTH0;
    cth = CTH0 * (WTH0 + Weff) * NF;
end else begin
    gth = 1.0; // set gth to some value to prevent a singular G matrix
    cth = 0.0;
end
```

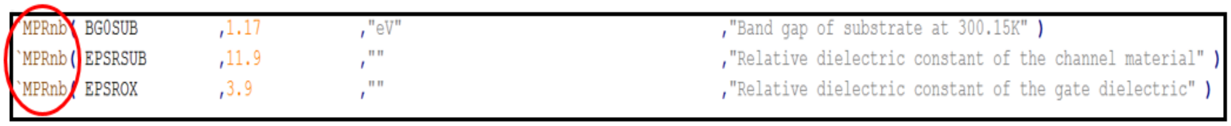
In BSIM-BULK106.2.0



```
// Parameters for self-heating
if((SHMOD != 0) && (RTH0 > 0.0) && (Weff_SH > 0.0)) begin
    gth = Weff_SH * NF / RTH0;
    cth = CTH0 * Weff_SH * NF;
end else begin
    // set gth to some value to prevent a singular G matrix
    gth = 1.0;
    cth = 0.0;
end
```

13. Changes in the macros definition

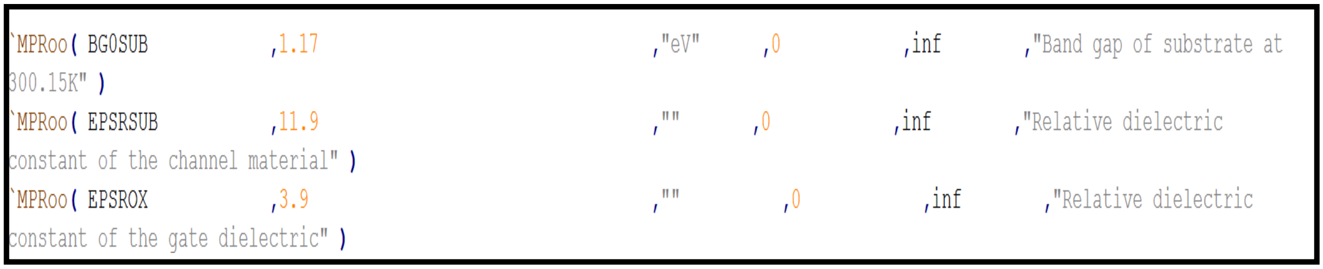
In BSIM6.1.1



```
`MPRnb( BG0SUB      ,1.17      ,"eV"                    ,"Band gap of substrate at 300.15K" )
`MPRnb( EPSRSUB     ,11.9      ,""                      ,"Relative dielectric constant of the channel material" )
`MPRnb( EPSROX      ,3.9       ,""                      ,"Relative dielectric constant of the gate dielectric" )
```

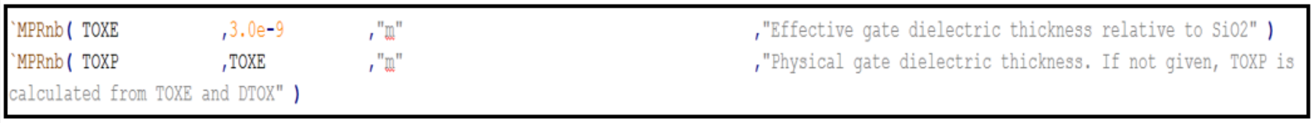
MPRnb -> MPRoz, Restrict their ranges to (0:inf)

In BSIM-BULK106.2.0



```
`MPRoo( BG0SUB          ,1.17                         ,"eV"     ,0          ,inf        ,"Band gap of substrate at
300.15K" )
`MPRoo( EPSRSUB         ,11.9                         ,""      ,0          ,inf        ,"Relative dielectric
constant of the channel material" )
`MPRoo( EPSROX          ,3.9                          ,""         ,0          ,inf        ,"Relative dielectric
constant of the gate dielectric" )
```

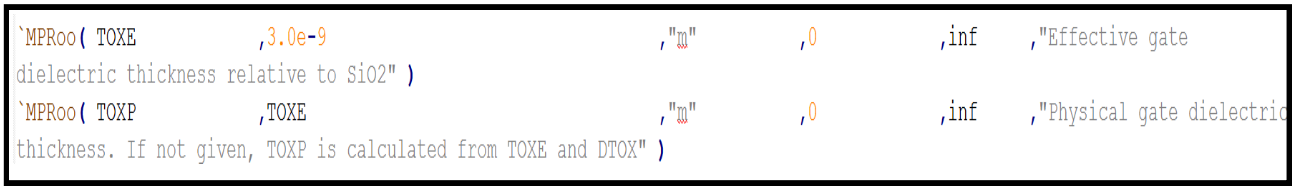
In BSIM6.1.1



```
`MPRnb( TOXE           ,3.0e-9       ,"m"                         ,"Effective gate dielectric thickness relative to SiO2" )
`MPRnb( TOXP           ,TOXE         ,"m"                         ,"Physical gate dielectric thickness. If not given, TOXP is
calculated from TOXE and DTOX" )
```

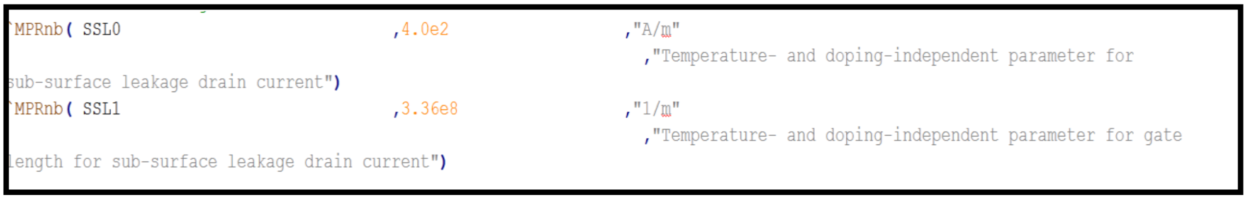
TOXE and TOXP must be positive, so MPRnb -> MPRoz, Restrict their ranges to (0:inf)

In BSIM-BULK106.2.0



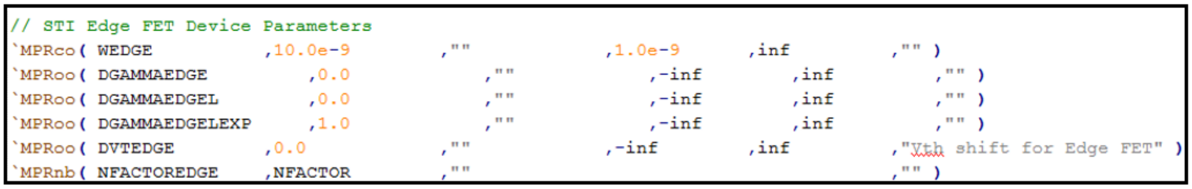
```
`MPRoo( TOXE          ,3.0e-9                              ,"m"        ,0            ,inf     ,"Effective gate
dielectric thickness relative to SiO2" )
`MPRoo( TOXP          ,TOXE                                ,"m"        ,0            ,inf     ,"Physical gate dielectric
thickness. If not given, TOXP is calculated from TOXE and DTOX" )
```

In BSIM-BULK106.2.0



```
`MPRnb( SSL0                        ,4.0e2                       ,"A/m"
                                                                   ,"Temperature- and doping-independent parameter for
sub-surface leakage drain current")
`MPRnb( SSL1                        ,3.36e8                      ,"1/m"
                                                                   ,"Temperature- and doping-independent parameter for gate
length for sub-surface leakage drain current")
```

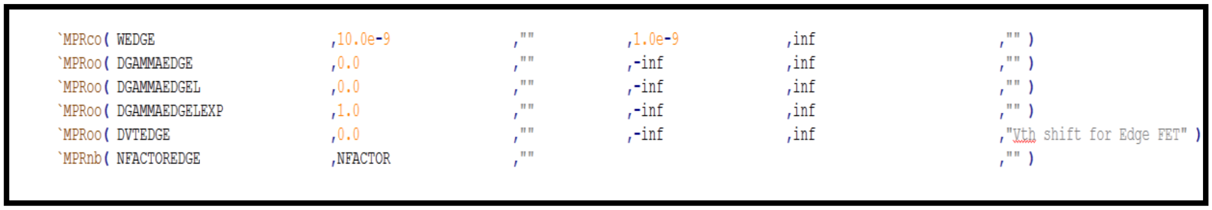
In BSIM6.1.1



```
// STI Edge FET Device Parameters
`MPRco( WEDGE          ,10.0e-9         ,""             ,1.0e-9       ,inf          ,"" )
`MPRoo( DGAMMAEDGE        ,0.0             ,""             ,-inf         ,inf          ,"" )
`MPRoo( DGAMMAEDGEL       ,0.0             ,""             ,-inf         ,inf          ,"" )
`MPRoo( DGAMMAEDGELEXP    ,1.0             ,""             ,-inf         ,inf          ,"" )
`MPRoo( DVTEDGE        ,0.0             ,""             ,-inf         ,inf          ,"Vth shift for Edge FET" )
`MPRnb( NFACTOREDGE    ,NFACTOR         ,""                                         ,"" )
```

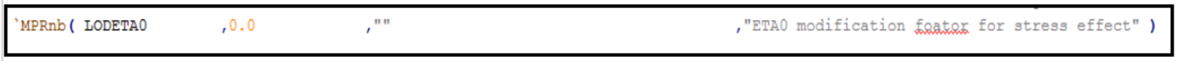
Parameter declarations aren't aligned properly

In BSIM-BULK106.2.0



```
    `MPRco( WEDGE                ,10.0e-9            ,""          ,1.0e-9       ,inf                  ,"" )
    `MPRoo( DGAMMAEDGE           ,0.0                ,""          ,-inf         ,inf                  ,"" )
    `MPRoo( DGAMMAEDGEL          ,0.0                ,""          ,-inf         ,inf                  ,"" )
    `MPRoo( DGAMMAEDGELEXP       ,1.0                ,""          ,-inf         ,inf                  ,"" )
    `MPRoo( DVTEDGE              ,0.0                ,""          ,-inf         ,inf                  ,"Vth shift for Edge FET" )
    `MPRnb( NFACTOREDGE          ,NFACTOR            ,""                                              ,"" )
```

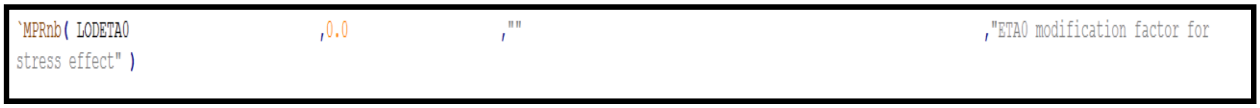
In BSIM6.1.1



```
`MPRnb( LODETA0          ,0.0           ,""                          ,"ETA0 modification foator for stress effect" )
```

"foator " => "factor"

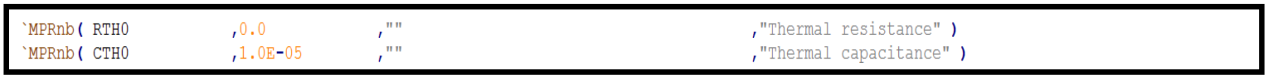
In BSIM-BULK106.2.0



```
`MPRnb( LODETA0                  ,0.0              ,""                                ,"ETA0 modification factor for
stress effect" )
```

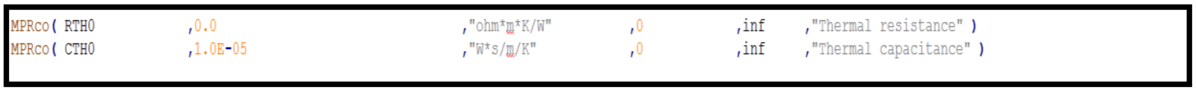
RTH0 and CTH0 must be positive

In BSIM6.1.1



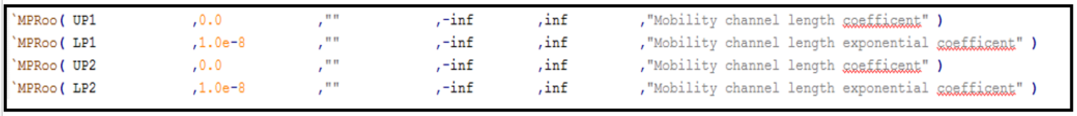
```
`MPRnb( RTH0          ,0.0           ,""                          ,"Thermal resistance" )
`MPRnb( CTH0          ,1.0E-05       ,""                          ,"Thermal capacitance" )
```

In BSIM-BULK106.2.0



```
MPRco( RTH0          ,0.0                          ,"ohm*m*K/W"     ,0          ,inf     ,"Thermal resistance" )
MPRco( CTH0          ,1.0E-05                      ,"W*s/m/K"       ,0          ,inf     ,"Thermal capacitance" )
```

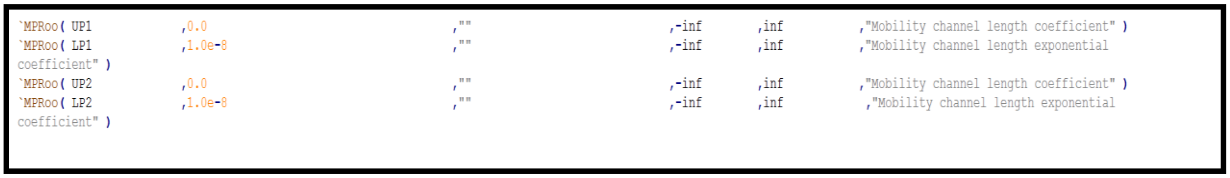
In BSIM6.1.1



```
`MPRoo( UP1          ,0.0          ,""           ,-inf        ,inf          ,"Mobility channel length coefficent" )
`MPRoo( LP1          ,1.0e-8       ,""           ,-inf        ,inf          ,"Mobility channel length exponential coefficent" )
`MPRoo( UP2          ,0.0          ,""           ,-inf        ,inf          ,"Mobility channel length coefficent" )
`MPRoo( LP2          ,1.0e-8       ,""           ,-inf        ,inf          ,"Mobility channel length exponential coefficent" )
```

"coefficent " => "coefficient"

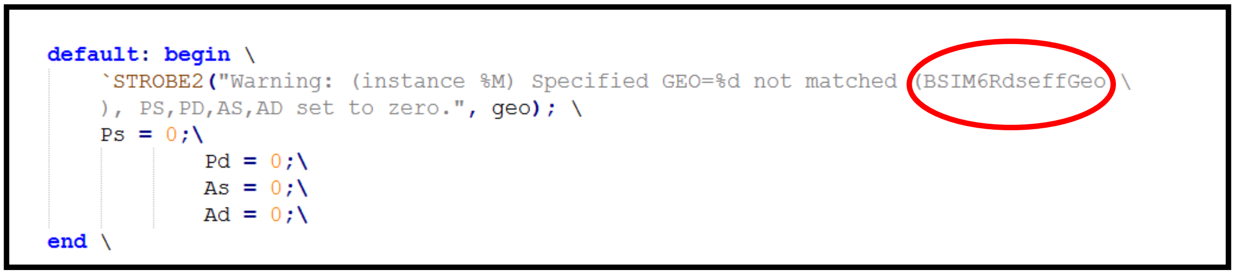
In BSIM-BULK106.2.0



```
`MPRoo( UP1          ,0.0                    ,""                 ,-inf      ,inf         ,"Mobility channel length coefficient" )
`MPRoo( LP1          ,1.0e-8                 ,""                 ,-inf      ,inf         ,"Mobility channel length exponential
coefficient" )
`MPRoo( UP2          ,0.0                    ,""                 ,-inf      ,inf         ,"Mobility channel length coefficient" )
`MPRoo( LP2          ,1.0e-8                 ,""                 ,-inf      ,inf          ,"Mobility channel length exponential
coefficient" )
```

14. Other Changes

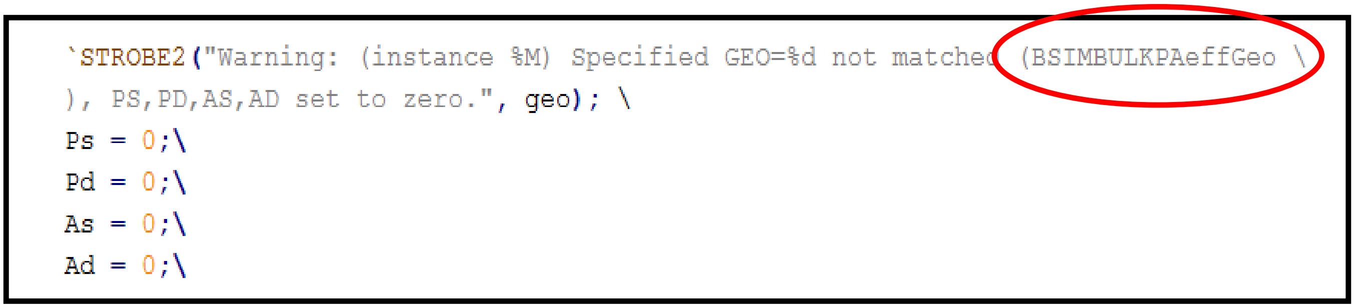
In BSIM6.1.1



```
default: begin \
    `STROBE2("Warning: (instance %M) Specified GEO=%d not matched (BSIM6RdseffGeo \
    ), PS,PD,AS,AD set to zero.", geo); \
    Ps = 0;\
            Pd = 0;\
            As = 0;\
            Ad = 0;\
end \
```

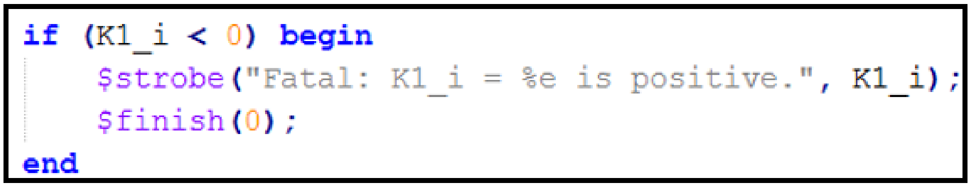
A TYPO ERROR

In BSIM-BULK106.2.0



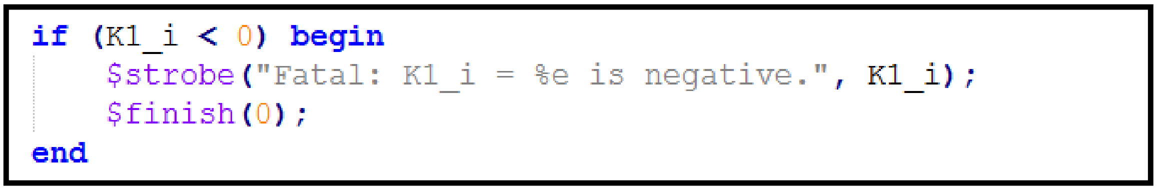
```
`STROBE2("Warning: (instance %M) Specified GEO=%d not matched (BSIMBULKPAeffGeo \
), PS,PD,AS,AD set to zero.", geo); \
Ps = 0;\
Pd = 0;\
As = 0;\
Ad = 0;\
```

In BSIM6.1.1



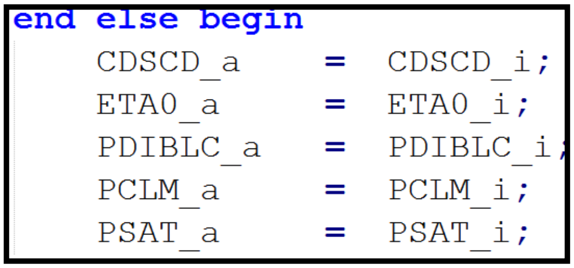
```
if (K1_i < 0) begin
    $strobe("Fatal: K1_i = %e is positive.", K1_i);
    $finish(0);
end
```

In BSIM-BULK106.2.0



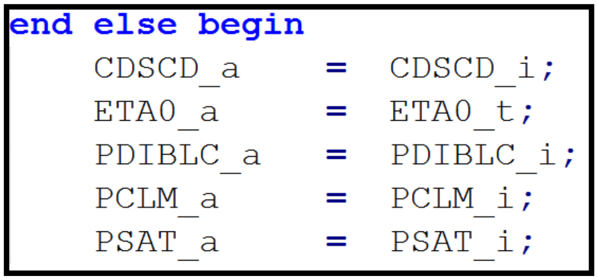
```
if (K1_i < 0) begin
    $strobe("Fatal: K1_i = %e is negative.", K1_i);
    $finish(0);
end
```

In BSIM6.1.1



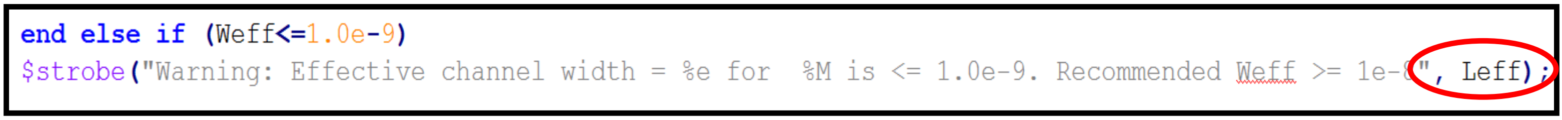
```
end else begin
    CDSCD_a     =  CDSCD_i;
    ETA0_a      =  ETA0_i;
    PDIBLC_a    =  PDIBLC_i;
    PCLM_a      =  PCLM_i;
    PSAT_a      =  PSAT_i;
```

In BSIM-BULK106.2.0



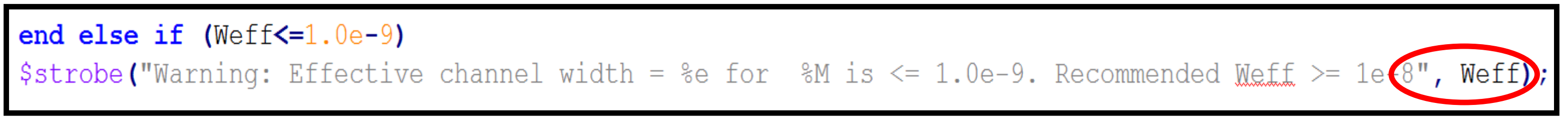
```
end else begin
    CDSCD_a     =  CDSCD_i;
    ETA0_a      =  ETA0_t;
    PDIBLC_a    =  PDIBLC_i;
    PCLM_a      =  PCLM_i;
    PSAT_a      =  PSAT_i;
```

In BSIM6.1.1



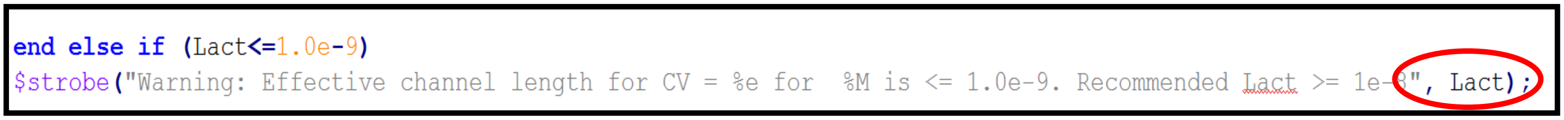
```
end else if (Weff<=1.0e-9)
$strobe("Warning: Effective channel width = %e for  %M is <= 1.0e-9. Recommended Weff >= 1e-8", Leff);
```

In BSIM-BULK106.2.0



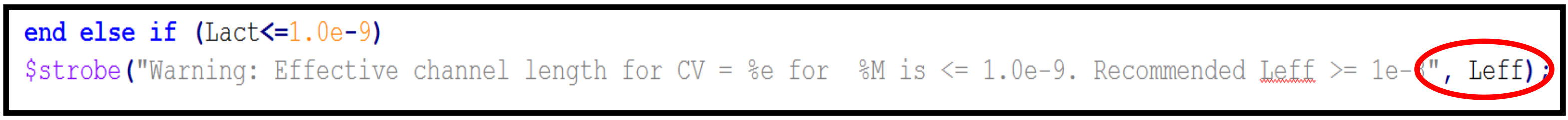
```
end else if (Weff<=1.0e-9)
$strobe("Warning: Effective channel width = %e for  %M is <= 1.0e-9. Recommended Weff >= 1e-8", Weff);
```

In BSIM6.1.1



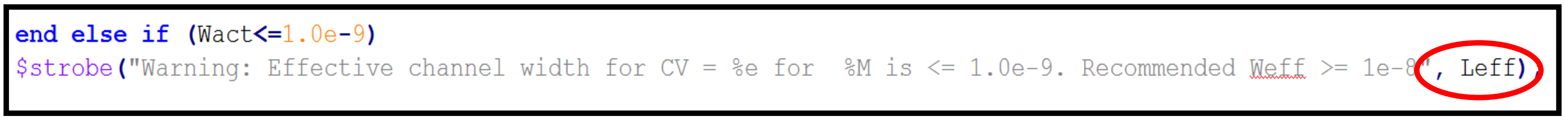
```
end else if (Lact<=1.0e-9)
$strobe("Warning: Effective channel length for CV = %e for  %M is <= 1.0e-9. Recommended Lact >= 1e-8", Lact);
```

In BSIM-BULK106.2.0



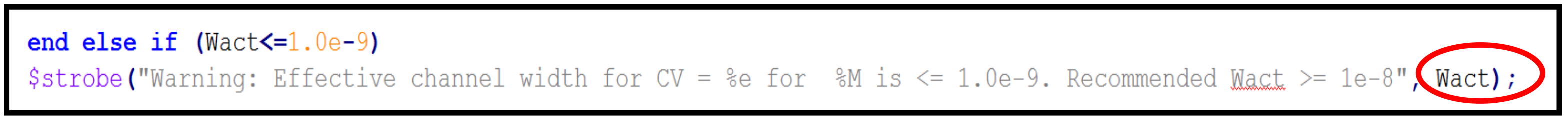
```
end else if (Lact<=1.0e-9)
$strobe("Warning: Effective channel length for CV = %e for  %M is <= 1.0e-9. Recommended Leff >= 1e-8", Leff);
```

In BSIM6.1.1



```
end else if (Wact<=1.0e-9)
$strobe("Warning: Effective channel width for CV = %e for  %M is <= 1.0e-9. Recommended Weff >= 1e-8", Leff);
```

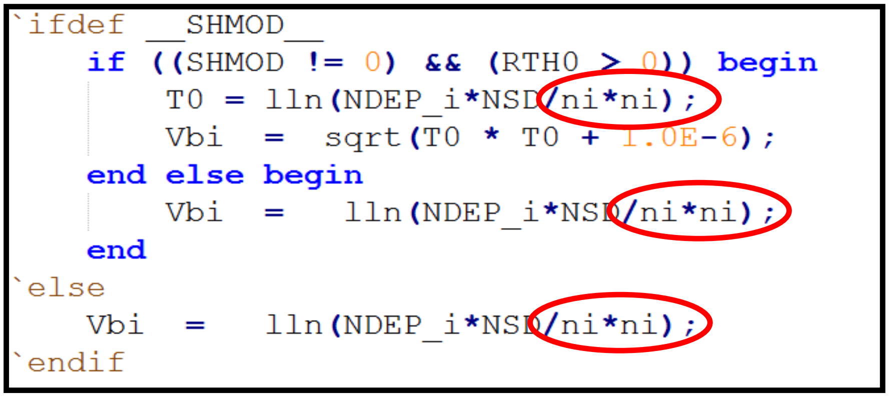
In BSIM-BULK106.2.0



```
end else if (Wact<=1.0e-9)
$strobe("Warning: Effective channel width for CV = %e for  %M is <= 1.0e-9. Recommended Wact >= 1e-8", Wact);
```

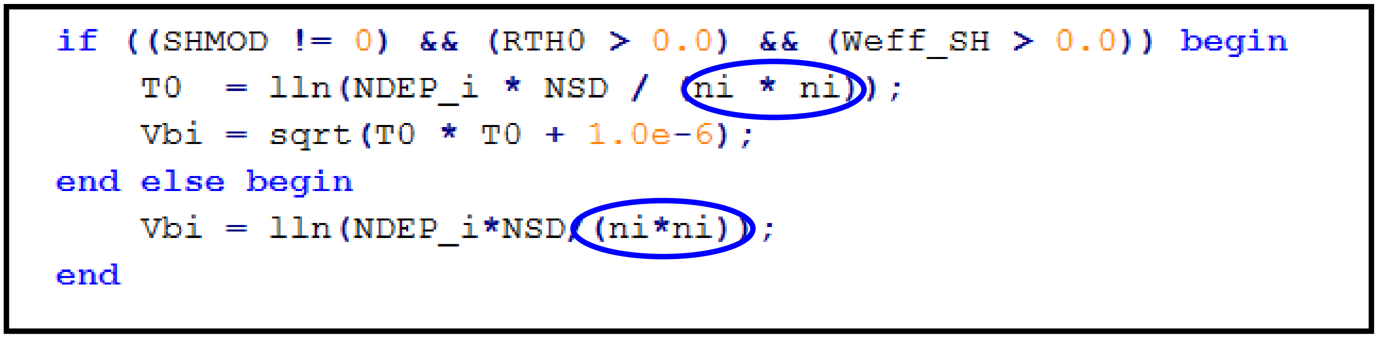
15. Modified the expression of built-in potential

In BSIM6.1.1



```
`ifdef __SHMOD__
    if ((SHMOD != 0) && (RTH0 > 0)) begin
        T0 = lln(NDEP_i*NSD/ni*ni);
        Vbi  =  sqrt(T0 * T0 + 1.0E-6);
    end else begin
        Vbi  =   lln(NDEP_i*NSD/ni*ni);
    end
`else
    Vbi  =   lln(NDEP_i*NSD/ni*ni);
`endif
```

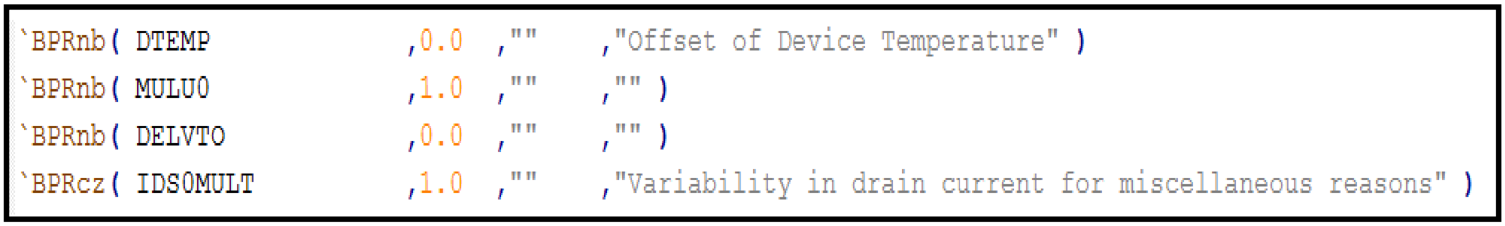
In BSIM-BULK106.2.0



```
if ((SHMOD != 0) && (RTH0 > 0.0) && (Weff_SH > 0.0)) begin
    T0  = lln(NDEP_i * NSD / (ni * ni));
    Vbi = sqrt(T0 * T0 + 1.0e-6);
end else begin
    Vbi = lln(NDEP_i*NSD/(ni*ni));
end
```

16. DTEMP, MULU0, DELVTO and IDS0MULT have modified to both instance and model parameters

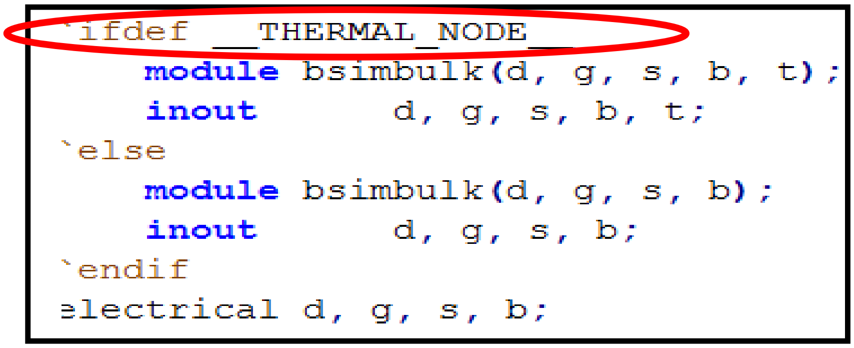
In BSIM-BULK106.2.0



```
`BPRnb( DTEMP            ,0.0  ,""    ,"Offset of Device Temperature" )
`BPRnb( MULU0            ,1.0  ,""    ,"" )
`BPRnb( DELVTO           ,0.0  ,""    ,"" )
`BPRcz( IDS0MULT         ,1.0  ,""    ,"Variability in drain current for miscellaneous reasons" )
```

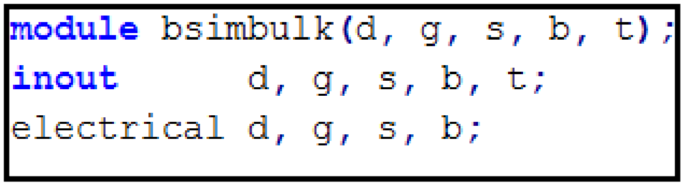
17. Removed all the "ifdef" statements from the code.

In BSIM6.1.1



```
`ifdef __THERMAL_NODE__
    module bsimbulk(d, g, s, b, t);
    inout       d, g, s, b, t;
`else
    module bsimbulk(d, g, s, b);
    inout       d, g, s, b;
`endif
electrical d, g, s, b;
```

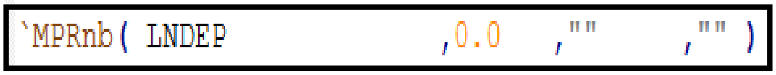
In BSIM-BULK106.2.0



```
module bsimbulk(d, g, s, b, t);
inout       d, g, s, b, t;
electrical d, g, s, b;
```

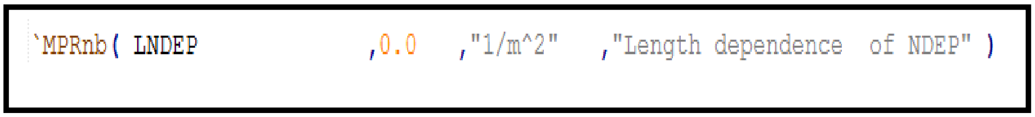
18. Added units and descriptions for all the parameters.

In BSIM6.1.1



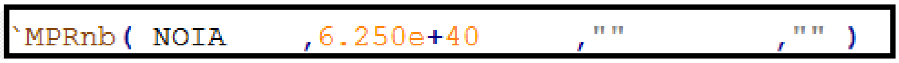
```
`MPRnb( LNDEP                ,0.0    ,""       ,"" )
```

In BSIM-BULK106.2.0



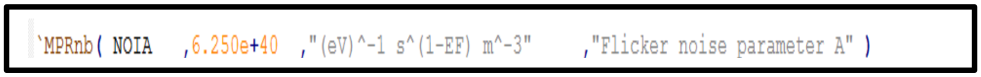
```
`MPRnb( LNDEP                ,0.0    ,"1/m^2"    ,"Length dependence  of NDEP" )
```

In BSIM6.1.1



```
`MPRnb( NOIA      ,6.250e+40       ,""              ,"" )
```

In BSIM-BULK106.2.0



```
`MPRnb( NOIA   ,6.250e+40  ,"(eV)^-1 s^(1-EF) m^-3"     ,"Flicker noise parameter A" )
```

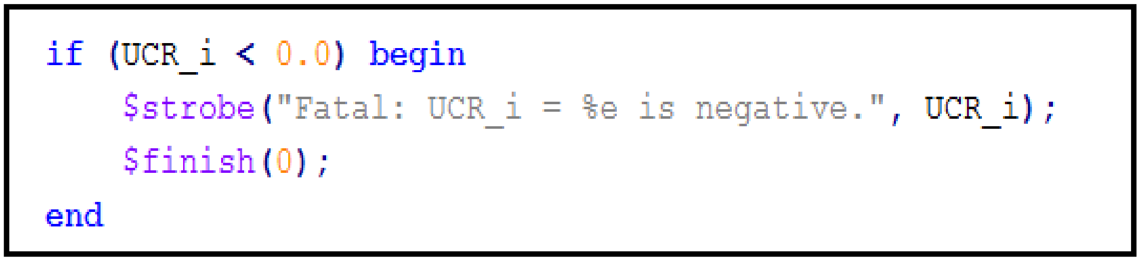
19. Added modified function of VDSX to avoid negative GDS issue

In BSIM-BULK106.2.0

$$\boxed{Vdsx = \frac{2}{AVDSX}\ln(1+e^{AVDSX.V_{DS}}) - V_{DS} - \frac{2}{AVDSX}\ln(2)}$$

20. Removed clamping from UCR

In BSIM6.1.1



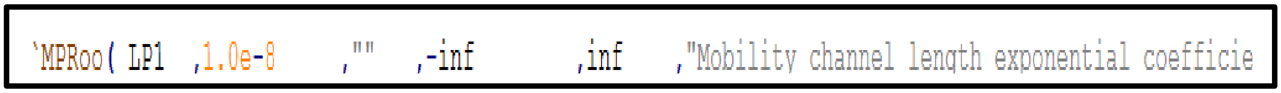
```
if (UCR_i < 0.0) begin
    $strobe("Fatal: UCR_i = %e is negative.", UCR_i);
    $finish(0);
end
```

In BSIM-BULK106.2.0

Clamping of UCR has been removed

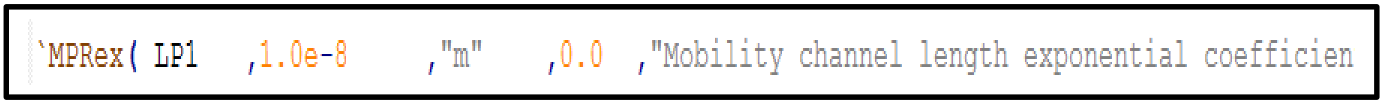
21. Added protection to the following parameters: LP1, LP2, NJS, NJD, and XJBVD.

In BSIM6.1.1



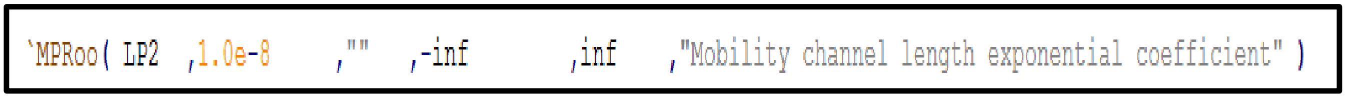
```
`MPRoo( LP1  ,1.0e-8     ,""   ,-inf       ,inf    ,"Mobility channel length exponential coefficie
```

In BSIM-BULK106.2.0



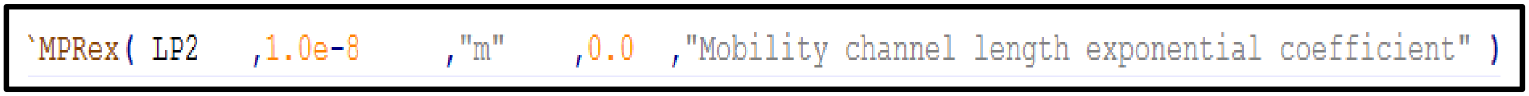
```
`MPRex( LP1   ,1.0e-8      ,"m"     ,0.0   ,"Mobility channel length exponential coefficien
```

In BSIM6.1.1



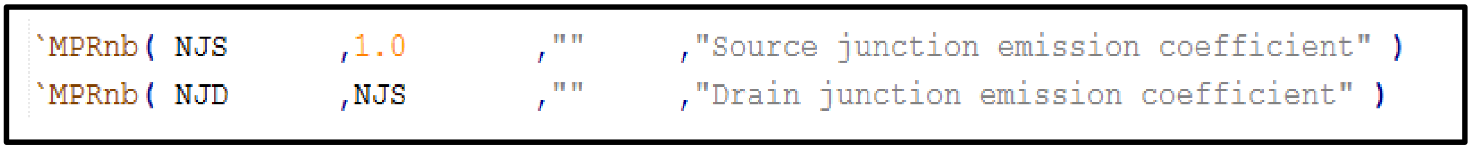
```
`MPRoo( LP2  ,1.0e-8     ,""   ,-inf       ,inf     ,"Mobility channel length exponential coefficient" )
```

In BSIM-BULK106.2.0



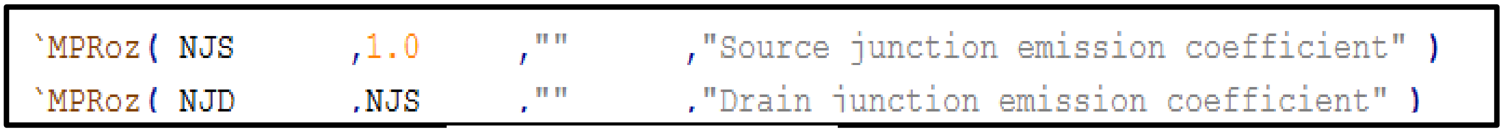
```
`MPRex( LP2    ,1.0e-8      ,"m"     ,0.0   ,"Mobility channel length exponential coefficient" )
```

In BSIM6.1.1



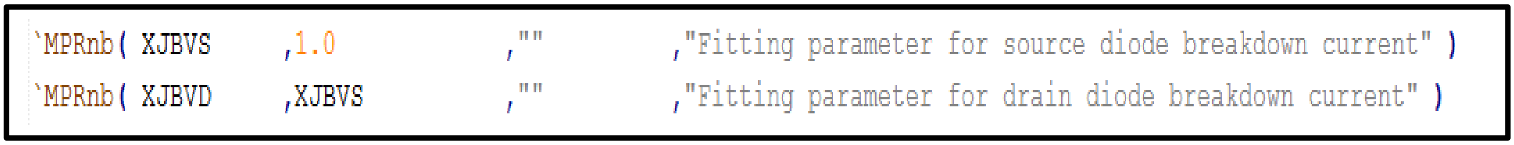
```
`MPRnb( NJS        ,1.0          ,""        ,"Source junction emission coefficient" )
`MPRnb( NJD        ,NJS          ,""        ,"Drain junction emission coefficient" )
```

In BSIM-BULK106.2.0



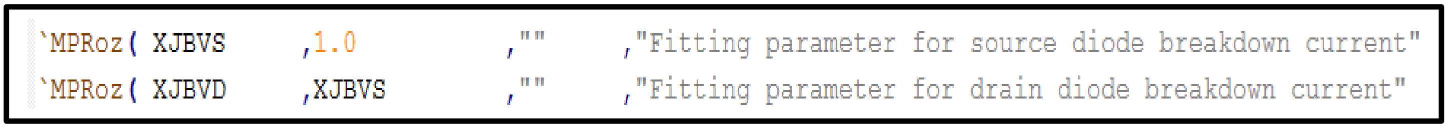
```
`MPRoz( NJS        ,1.0        ,""        ,"Source junction emission coefficient" )
`MPRoz( NJD        ,NJS        ,""        ,"Drain junction emission coefficient" )
```

In BSIM6.1.1



```
`MPRnb( XJBVS      ,1.0             ,""          ,"Fitting parameter for source diode breakdown current" )
`MPRnb( XJBVD      ,XJBVS           ,""          ,"Fitting parameter for drain diode breakdown current" )
```

In BSIM-BULK106.2.0



```
`MPRoz( XJBVS      ,1.0            ,""      ,"Fitting parameter for source diode breakdown current"
`MPRoz( XJBVD      ,XJBVS          ,""      ,"Fitting parameter for drain diode breakdown current"
```