Summary of changes done:

1. **2013bug11:** Added identifying names to all noise sources
2. **2013enh5:** Reviewed possibilities for collapsed S/D in the Verilog-A code. RDSMOD=2 has been proposed as a solution to this. Thus the solution is similar to 2015enh1.
3. **2014bug2:** Node Collapsing Difference between C code and Verilog-A code for SOIMOD=2
4. **2014bug6:** Used binning parameter for Npeak.
5. **2015enh1:** Improved implementation of source/drain resistance when RSH=0 to speed up simulation.
6. **2015enh2:** Introduced parameters for better fitting accuracy when using MTRLMOD=0 (body contacted device).
7. **2016bug1:** Corrected IGIDL Calculation.
8. **2016bug2:** When pParam_B4SOIdlcig<0, clamp it to zero and issue a warning message.
9. **2016bug3:** Used Binning parameter **pParam_B4SOInpeak** for **B4SOInpeak** at all places.
10. **2016bug4:** Corrected RDSMOD Calculation for hidden states.
11. **2016bug5:** Removed redundant variable VTH_CV.
12. **2016bug6:** Limited NOFF and NOFF2 to values > 0
13. **2016bug7:** Use *lln* only when required, else use *ln* as before
14. **2016bug8:** Restored NF back to integer.
15. **2016bug9:** Separated issuing warning (only when NODECHK=1) and grounding nodes (irrespective of NODECHK)
16. **2016bug10:** Corrected the incorrect behavior with regard to SHMOD in Spectre simulations.
17. **2016enh1:** Use of Clamped log function
18. **2016enh2:** Updated Copyright Statement
19. **2016enh3:** Proper units and ranges set for parameters
20. **2016enh4:** Improved the clamped log function efficiency
21. **2016enh5:** Modified gate charge and capacitance calculations to include overlap charge
22. **2017enh1:** Binning of Built-In Potential Lowering (ΔVbi) Model Parameters.
23. **2017enh2:** Output variables added for self-heating mode**.**

---

1. 2013bug11: Added identifying names to all noise sources.

1)

Old Code	*begin*
	*I(b, p) <+ B4SOItype * Ibp;*
	*I(b, p) <+ white_noise(fourkt*abs(Ibp)/(abs(vbp)+1.0e-9));*
	end

New Code	begin
	I(b, p) <+ B4SOItype * Ibp;
	I(b, p) <+ white_noise(fourkt*abs(Ibp)/(abs(vbp)+1.0e-9), "rbp");
	end

2)

Old Code	*I(di,si) <+ white_noise(sid*abs(1.0-ctnoi * ctnoi));*
	*I(N) <+ V(N) * sf * SCALEN;*
	*I(N) <+ white_noise(sid/(sf*sf*SCALEN*SCALEN));*

New Code	I(di,si) <+ white_noise(sid*abs(1.0-ctnoi * ctnoi), "id");
	I(N) <+ V(N) * sf * SCALEN;
	I(N) <+ white_noise(sid/(sf*sf*SCALEN*SCALEN), "corl");

2. RDSMOD (2013enh5, 2015enh1)

RDSMOD=0: Bias independent external, dependent internal.
RDSMOD=1: Both bias independent and dependent are external.
RDSMOD=2: Both bias independent and dependent are internal.

1) The following Lines were removed (v4.5) (Redundant)

Line no.1529	real B4SOIsourceConductance;
Line no.1530	real B4SOIdrainConductance;

2)

Old Code	*if (B4SOIrdsMod) begin /* v4.0 */ // {*
	*PowWeffWr = pParam_B4SOIrds0denom * B4SOInf;*

New Code	if (B4SOIrdsMod == 1) begin /* v4.0 */ // { /* 2013.enh5, 2015.enh1 */ /* v4.6 */
	PowWeffWr = pParam_B4SOIrds0denom * B4SOInf;

3)

```
/* process source/drain series resistance */

     B4SOIdrainResistance = B4SOIsheetResistance * B4SOIdrainSquares;
     if (B4SOIdrainResistance <= 1.0e-3)          /* 2013.enh5, 2015.enh1 */ /* v4.6 */
               B4SOIdrainResistance = 1.0e-3;

     B4SOIsourceResistance = B4SOIsheetResistance * B4SOIsourceSquares;
     if (B4SOIsourceResistance <= 1.0e-3)         /* 2013.enh5, 2015.enh1 */ /* v4.6 */
               B4SOIsourceResistance = 1.0e-3;
```

4)

Old Code	*if (!B4SOIrdsMod) begin*
	*rds0 = (pParam_B4SOIrdsw + pParam_B4SOIprt * T3)*
	/ pParam_B4SOIrds0denom;
	end else begin / v4.0 */*
	*PowWeffWr = pParam_B4SOIrds0denom * B4SOInf;*
	*T10 = pParam_B4SOIprt * T3;*

New Code	if (B4SOIrdsMod != 1) begin /* 2013.enh5, 2015.enh1 */ /* v4.6 */
	rds0 = (pParam_B4SOIrdsw + pParam_B4SOIprt * T3)
	/ pParam_B4SOIrds0denom;
	rd0 = 0;
	rs0 = 0;
	end else begin /* v4.0 */ /* 2013.enh5, 2015.enh1 */ /* v4.6 */
	rds0 = 0;
	PowWeffWr = pParam_B4SOIrds0denom * B4SOInf;
	T10 = pParam_B4SOIprt * T3;

5)

```
if (B4SOIrdsMod == 2) begin
    Rds = B4SOIdrainResistance + Rds + B4SOIsourceResistance;
end
```

6)

```
/* v4.0 Calculate bias-dependent external S/D resistance */
   if (B4SOIrdsMod == 1) begin          /* 2013.enh5, 2015.enh1 */ /* Updated in v4.6 */
       Rds = 0.0;
       /* Rs(V) */
       T0 = vgs - pParam_B4SOIvfbsd;
       T1 = sqrt(T0 * T0 + 1.0e-4);
       vgs_eff = 0.5 * (T0 + T1);

       T0 = 1.0 + pParam_B4SOIprwg * vgs_eff;
       T1 = -pParam_B4SOIprwb * vbs;
       T2 = 1.0 / T0 + T1;
       T3 = T2 + sqrt(T2 * T2 + 0.01);

       T4 = rs0 * 0.5;
       Rs = rswmin + T3 * T4 + B4SOIsourceResistance;


       /* Rd(V) */
       T0 = vgd - pParam_B4SOIvfbsd;
       T1 = sqrt(T0 * T0 + 1.0e-4);
       vgd_eff = 0.5 * (T0 + T1);

       T0 = 1.0 + pParam_B4SOIprwg * vgd_eff;
       T1 = -pParam_B4SOIprwb * vbd;

       T2 = 1.0 / T0 + T1;
       T3 = T2 + sqrt(T2 * T2 + 0.01);
       T4 = rd0 * 0.5;
       Rd = rdwmin + T3 * T4 + B4SOIdrainResistance;

   end else begin
       Rs = B4SOIsourceResistance;
       Rd = B4SOIdrainResistance;
    end
   if (B4SOIrdsMod == 2) begin   /* 2013.enh5, 2015.enh1 */ /* v4.6 */
       Rs = 0;
       Rd = 0;
   end
   if (B4SOIrdsMod != 2) begin          /* 2013.enh5, 2015.enh1 */ /* v4.6 */
       B4SOIgstot = 1 / Rs ;
       B4SOIgdtot = 1 / Rd ;
   end
```

7)

```
if (B4SOIrdsMod != 2) begin           /* 2013.enh5, 2015.enh1 */ /* v4.6 */
    I(d, di) <+ V(d,di) * B4SOIgdtot;
    I(d, di) <+ white_noise( abs(fourkt * gdnoise), "rd");
```

```
                I(s, si) <+ V(s,si) * B4SOIgstot;
                I(s, si) <+ white_noise( abs(fourkt * gsnoise), "rs");
        end else begin
                V(d, di) <+ 0.0;
                V(s, si) <+ 0.0;
        end
```

3. 2014bug2: Node Collapsing Difference between C code and Verilog-A code for SOIMOD=2

Line no. 234: TNODEOUT changed from model parameter to instance parameter.
Line no. 266: Use of model parameter NODECHK flag.
Line no. 2457-2515: “NODECHK” warnings.
Line no. 4230-4251: Calculate temperature dependent values for self-heating effect.
Line no. 7916-7947: Self-Heating Network.

4. 2014bug6: “NPEAK”

Old Code	*pParam_B4SOIqsi = `Charge_q * B4SOInpeak* ** (1.0 + pParam_B4SOIlpe0 / pParam_B4SOIleff)* ** 1e6 * B4SOItsi;*

New Code	pParam_B4SOIqsi = `Charge_q * pParam_B4SOInpeak
	* (1.0 + pParam_B4SOIlpe0 / pParam_B4SOIleff)
	* 1e6 * B4SOItsi;

6. 2015enh2: : Introduced parameters for better fitting accuracy when using MTRLMOD=0 (body contacted device).

a) NOFF2

1) NOFF2 was added

```
        `MPRnb( NOFF2        ,NOFF          ,""            ,"C-V turn-on/off parameter /* v4.6 " )
```

2) PNOFF2 was added

```
        `MPRnb( PNOFF2       ,PNOFF         ,""       ,"Cross-term dependence of noff2 /* v4.6 " )
```

3) "B4SOInoff2" was added

```
        real B4SOInoff2;
```

4) "pParam_B4SOInoff2" was added

```
         real pParam_B4SOInoff2;
```

5) "B4SOInoff2" was added

```
        B4SOInoff2 = NOFF2;
```

6) "noff2" was added for binning

```
        pParam_B4SOInoff2  = B4SOInoff2  + LNOFF2  * Inv_L + WNOFF2  * Inv_W + PNOFF2  *
                                Inv_LW;  /*  v4.6  */
```

7) Check for "noff2" was added

```
        if (pParam_B4SOInoff2 < 0.1)
                $strobe("Warning: Noff2 = %g is too small.", pParam_B4SOInoff2);
        else if (pParam_B4SOInoff2 > 4.0)
                $strobe("Warning: Noff2 = %g is too large.", pParam_B4SOInoff2);
```

8) "noff2" was added

```
        real noff2;        /* 2015enh2 */
```

9) "NOFF2" was added

```
        noff2 = n_CV * pParam_B4SOInoff2;    /* 2015enh2 */      /* v4.6 */
```

10)

Old Code	*if (B4SOIagbcp2 > 0) begin*
	*ExpVgst2 = ExpVgst * exp(-eggbcp2 / noff / Vtm);*
	*Vgsteff2 = noff * Vtm * ln(1.0 + ExpVgst2);*
	end

New Code	if (B4SOIagbcp2 > 0) begin
	ExpVgst2 = ExpVgst * exp(-eggbcp2 / noff2 / Vtm); /* 2015enh2 */ /* v4.6 */
	Vgsteff2 = noff2 * Vtm * ln(1.0 + ExpVgst2);
	end

11)

Old Code	*if (B4SOIagbcp2 > 0) begin*
	*ExpVgst2 = ExpVgst * exp(-eggbcp2 / noff / Vtm);*
	*Vgsteff2 = noff * Vtm * ln(1.0 + ExpVgst2);*
	end

New Code	if (B4SOIagbcp2 > 0) begin
	ExpVgst2 = ExpVgst * exp(-eggbcp2 / noff2 / Vtm); /* 2015enh2 */ /* v4.6 */
	Vgsteff2 = noff2 * Vtm * ln(1.0 + ExpVgst2);
	end

12) "T10" was added

```
        if (B4SOIagbcp2 > 0) begin
                T10 = noff2 * Vtm;                          /* 2015enh2 */      /* v4.6 */
        VgstNVt2 = pParam_B4SOImstarcv * (Vgst - pParam_B4SOIdelvt - eggbcp2) / T10;
```

13)

Old Code	*T0 = (Vgst - pParam_B4SOIdelvt - pParam_B4SOIvoffcv - eggbcp2) / (noff * Vtm);*
New Code	T0 = (Vgst - pParam_B4SOIdelvt - pParam_B4SOIvoffcv - eggbcp2) / (noff2 * Vtm); /* 2015enh2 */ /* v4.6 */

b) Replacement of constants with model parameter:

8. 2016bug2:

```
    if (pParam_B4SOIdlcig < 0.0) begin                  /* 2013.bug2 */ /* v4.6 */
       $strobe("Warning: pParam_B4SOIdlcig = %g is negative. Set to zero.", pParam_B4SOIdlcig);
       pParam_B4SOIdlcig = 0;
    end
```

9. 2016bug3:

Moved these lines after binning and used pParam_B4SOInpeak in place of B4SOInpeak.

Old Code

```
/* Check of Nch and Body thickness for SOIMOD = 2 */
if (B4SOIsoiMod == 2) begin
   if (B4SOImtrlMod) begin
      NchMax = (B4SOIbg0sub - 0.1) / `Charge_q * 2e-6 * epssub / (B4SOIetsi * B4SOIetsi);
      if (B4SOInpeak > NchMax ) begin
         $strobe("SOIMOD=2 cannot support given Nch=%g cm^-3 and Etsi=%g m
",B4SOInpeak,B4SOIetsi);
         $strobe("Exceeds maximum allowed band bending of (Eg-0.1)eV");
         $strobe("Nch is set to  %g cm^-3", NchMax);
         B4SOInpeak = NchMax;
      end
   end else begin
      NchMax = (1.12 - 0.1) / `Charge_q * 2e-6 * epssub / (B4SOItsi * B4SOItsi);
      if (B4SOInpeak > NchMax ) begin
         $strobe("SOIMOD=2 cannot support given Nch=%g cm^-3 and Tsi=%g m
",B4SOInpeak,B4SOItsi);
         $strobe("Exceeds maximum allowed band bending of (Eg-0.1)eV");
         $strobe("Nch is set to  %g cm^-3", NchMax);
         B4SOInpeak = NchMax;
      end
   end
end

/* process effective silicon film thickness */
B4SOIcbox = 3.453133e-11 / B4SOItbox;
if (B4SOImtrlMod) begin
   B4SOIcsi = `EPSSI / B4SOIetsi;
end else begin
   B4SOIcsi = `EPSSI / B4SOItsi;
end
// Cboxt = B4SOIcbox * B4SOIcsi / (B4SOIcbox + B4SOIcsi);
/* v3.2 */
if (B4SOImtrlMod) begin
   Qsi = `Charge_q * B4SOInpeak
         * (1.0 + B4SOIlpe0 / B4SOIl) * 1e6 * B4SOIetsi;
end else begin
   Qsi = `Charge_q * B4SOInpeak
         * (1.0 + B4SOIlpe0 / B4SOIl) * 1e6 * B4SOItsi;
end
Vbs0t = 0.8 - 0.5 * Qsi / B4SOIcsi + B4SOIvbsa;

if (B4SOIsoiMod == 3) /* auto selection */
```

```
  if (Vbs0t > B4SOIvbs0fd)
    B4SOIisoiMod = 2; /* ideal FD mode */
  else
    if (Vbs0t < B4SOIvbs0pd)
      B4SOIisoiMod = 0; /* BSIMPD */
    else
      B4SOIisoiMod = 1;
```

New Code

```
/* Check of Nch and Body thickness for SOIMOD = 2 */
if (B4SOIisoiMod == 2) begin                    /* 2016.bug3 */ /* v4.6 */
   if (B4SOImtrlMod) begin
      NchMax = (B4SOIbg0sub - 0.1) / `Charge_q * 2e-6 * epssub / (B4SOIetsi * B4SOIetsi);
      if (pParam_B4SOInpeak > NchMax ) begin
         $strobe("SOIMOD=2 cannot support given Nch=%g cm^-3 and Etsi=%g m",
                    pParam_B4SOInpeak,B4SOIetsi);
         $strobe("Exceeds maximum allowed band bending of (Eg-0.1)eV");
         $strobe("Nch is set to  %g cm^-3", NchMax);
         pParam_B4SOInpeak = NchMax;
      end
   end else begin
      NchMax = (1.12 - 0.1) / `Charge_q * 2e-6 * epssub / (B4SOItsi * B4SOItsi);
      if (pParam_B4SOInpeak > NchMax ) begin
         $strobe("SOIMOD=2 cannot support given Nch=%g cm^-3 and Tsi=%g m ",
                    pParam_B4SOInpeak,B4SOItsi);
         $strobe("Exceeds maximum allowed band bending of (Eg-0.1)eV");
         $strobe("Nch is set to  %g cm^-3", NchMax);
         pParam_B4SOInpeak = NchMax;
      end
   end
end

/* process effective silicon film thickness */
B4SOIcbox = 3.453133e-11 / B4SOItbox;
if (B4SOImtrlMod) begin
   B4SOIcsi = `EPSSI / B4SOIetsi;
end else begin
   B4SOIcsi = `EPSSI / B4SOItsi;
end
// Cboxt = B4SOIcbox * B4SOIcsi / (B4SOIcbox + B4SOIcsi);
/* v3.2 */
if (B4SOImtrlMod) begin
   Qsi = `Charge_q * pParam_B4SOInpeak
         * (1.0 + B4SOIlpe0 / B4SOIl) * 1e6 * B4SOIetsi;
end else begin
   Qsi = `Charge_q * pParam_B4SOInpeak
         * (1.0 + B4SOIlpe0 / B4SOIl) * 1e6 * B4SOItsi;
end
Vbs0t = 0.8 - 0.5 * Qsi / B4SOIcsi + B4SOIvbsa;
```

	if (B4SOIsoiMod == 3) /* auto selection */
	if (Vbs0t > B4SOIvbs0fd)
	B4SOIsoiMod = 2; /* ideal FD mode */
	else
	if (Vbs0t < B4SOIvbs0pd)
	B4SOIsoiMod = 0; /* BSIMPD */
	else
	B4SOIsoiMod = 1;

Old Code	if (B4SOInpeak > 1.0e20)
	$strobe("Warning: Npeak = %g is too large, should be less than 1.0e20, specified in cm^-3.",
	B4SOInpeak);

New Code	if (pParam_B4SOInpeak > 1.0e20) /* 2016.bug3 */ /* v4.6 */
	$strobe("Warning: Npeak = %g is too large, should be less than 1.0e20, specified in cm^-3.",
	pParam_B4SOInpeak);

10. 2016bug4:

Old Code	1) if (B4SOIrdsMod != 2) begin /* 2013.enh5, 2015.enh1 */ /* v4.6 */
	B4SOIgstot = 1 / Rs ;
	B4SOIgdtot = 1 / Rd ;
	end
	
	2)
	gdnoise = B4SOIgdtot;
	gsnoise = B4SOIgstot;
	
	3)
	if (B4SOIrdsMod != 2) begin /* 2013.enh5, 2015.enh1 */ /* v4.6 */
	I(d, di) <+ V(d,di) * B4SOIgdtot;
	I(d, di) <+ white_noise(abs(fourkt * gdnoise), "rd");
	I(s, si) <+ V(s,si) * B4SOIgstot;
	I(s, si) <+ white_noise(abs(fourkt * gsnoise), "rs");
	end else begin
	V(d, di) <+ 0.0;
	V(s, si) <+ 0.0;
	end

New Code	1)
	Variable B4SOIgdtot and B4SOIgstot were removed.
	
	2)
	
	if (B4SOIrdsMod != 2) begin /* 2016.bug4 */ /* v4.6 */
	gdnoise = 1 / Rd ;
	gsnoise = 1 / Rs ;
	end else begin
	gdnoise = 0.0;

	gsnoise = 0.0;
	end
	
	3)
	if (B4SOIrdsMod != 2) begin /* 2013.enh5, 2015.enh1, 2016.bug4 */ /* v4.6 */
	I(d, di) <+ V(d,di) / Rd ;
	I(d, di) <+ white_noise(abs(fourkt * gdnoise), "rd");
	I(s, si) <+ V(s,si) / Rs ;
	I(s, si) <+ white_noise(abs(fourkt * gsnoise), "rs");
	end else begin
	V(d, di) <+ 0.0;
	V(s, si) <+ 0.0;
	end

12. 2016bug6: NOFF and NOFF2

a) NOFF and NOFF2 were not prevented from being zero (Earlier there was only a warning if they were less than 0.1)

Fix: Limited NOFF and NOFF2 to values > 0

b) Included Vtm in noff and noff2 (Since the product noff * Vtm appears frequently)

Old Code	New code

`noff = n_CV * pParam_B4SOInoff;`
`noff2 = n_CV * pParam_B4SOInoff2;`
	`noff = n_CV * pParam_B4SOInoff * Vtm;`
	`noff2 = n_CV * pParam_B4SOInoff2 * Vtm;`

c) T10 is no longer used

(Earlier: T10 = noff * Vtm , T10 = noff2* Vtm)

13. 2016bug7: ln Vs lln

Switching from ***ln*** to ***lln*** (clamped log function) was unnecessary or redundant at some places, so in such instances, changed ***lln*** back to ***ln***

Example:

Old Code	`B4SOIcf = 2.0 * `EPSOX / `M_PI * lln(1.0 + 0.4e-6 / B4SOItox);`
New Code	`B4SOIcf = 2.0 * `EPSOX / `M_PI * ln(1.0 + 0.4e-6 / B4SOItox);`

14. 2016bug8: Restored NF back to integer. It had changed from IPIco to IPRco (integer to real) in beta2.

```
`IPIco( NF              ,1.0              ,""            ,1            ,inf        ,"Number of fingers" )
```

15. 2016bug9: If NODECHK=0 is specified, then not only are the warnings suppressed, but also the model behavior changes as the P and/or B nodes are not affected.

Fix: Separated issuing warnings (only when NODECHK=1) and grounding nodes (irrespective of NODECHK)

1. Added a block of lines (independent of NODECHK) just before we check for NODECHK (i.e. just above line #2415 in beta2 code) for grounding V(p) or V(b) as the case may be (but without issuing any warning)

2. The "*if (NODECHK)"* block is now being used only for the purpose of issuing warnings (and not for grounding any node)

OLD CODE (beta2)

```
`ifdef PORT_CONNECTED
   if (B4SOIsoiMod == 2) begin /* v3.2 */
      if (NODECHK) begin      /* 2014bug2 */
         if (SHMOD == 0) begin
            if (TNODEOUT == 0) begin //TNODEOUT = 0 starts here.
               if ($port_connected(p) && $port_connected(b)) begin
                  $strobe("Warning: 'P' and 'B' node were specified, however 'SOIMOD = 2' does not support 'P' or 'B'
nodes. Internally these nodes are tied to global ground.");
                  V(b) <+ 0;
               end else begin
                  if ($port_connected(p)) begin
                     $strobe ("Warning: 'P' node was specified, however 'SOIMOD = 2' does not support 'P' node.
Internally this node is tied to global ground.");
                     V(p) <+ 0;
                  end
               end
            end else begin //TNODEOUT = 0 for SHMOD = 0 ends here.
               if ($port_connected(p) && $port_connected(b) && $port_connected(t)) begin
                  $strobe("Warning: 'P' and 'B' node were specified, however 'SOIMOD = 2' does not support 'P' or 'B'
nodes. Internally these nodes are tied to global ground.");
                  V(b) <+ 0;
               end else begin
                  if ($port_connected(p) && $port_connected(b)) begin
                     $strobe ("Warning: 'P' node was specified, however 'SOIMOD = 2' does not support 'P' node.
Internally this node is tied to global ground.");
                  end
               end
            end         //TNODEOUT = 1 for SHMOD = 0 ends here.
         end else begin     //SHMOD = 0 ends here.
            if (TNODEOUT == 0) begin
               if ($port_connected(p) && $port_connected(b) && $port_connected(t)) begin
                  $strobe("Warning: 'P' and 'B' node were specified, however 'SOIMOD = 2' does not support 'P' or 'B'
nodes. Internally these nodes are tied to global ground.");
                  V(b) <+ 0;
               end else begin
                  if ($port_connected(p) && $port_connected(b)) begin
                     $strobe("Warning: 'P' and 'B' node were specified, however 'SOIMOD = 2' does not support 'P' or
'B' nodes. Internally these nodes are tied to global ground.");
                     V(b) <+ 0;
                  end else begin
                     if ($port_connected(p)) begin
                        $strobe ("Warning: 'P' node was specified however 'SOIMOD = 2' does not support 'P' node.
Internally this node is tied to global ground.");
                        V(p) <+ 0;
                     end
                  end
               end
            end else begin //TNODEOUT = 0 for SHMOD = 1 ends here.
```

```
               if ($port_connected(p) && $port_connected(b) && $port_connected(t)) begin
                  $strobe("Warning: 'P' and 'B' node were specified, however 'SOIMOD = 2' does not support 'P' or 'B'
nodes. Internally these nodes are tied to global ground.");
                  V(b) <+ 0;
               end else begin
                  if ($port_connected(p) && $port_connected(b)) begin
                     $strobe("Warning: 'P' and 'B' nodes were specified when TNODEOUT = 1. The 'B' node will be
treated as the 'T' (temperature) node, and internally the 'P' node will be tied to global ground.");
                     V(p) <+ 0;
                  end else begin
                     if ($port_connected(p)) begin
                        $strobe ("Warning: 'P' node has been specified with TNODEOUT = 1. The 'P' node will be treated
as the 'T' (temperature) node.");
                     end else begin
                        if (!($port_connected(p) && $port_connected(b) && $port_connected(t))) begin
                           $strobe("Warning: TNODEOUT = 1 requesting an external temperature node, but a fifth node
temperature is not supplied.");
                        end
                     end
                  end
               end
            end         //TNODEOUT = 1 for SHMOD = 1 ends here.
         end            //SHMOD = 1 ends here.
      end               //NODECHK ends here.
```

NEW CODE (beta3)

```
`ifdef PORT_CONNECTED
   if (B4SOIisoiMod == 2) begin /* v3.2 */
      if (SHMOD == 0) begin
         if (TNODEOUT == 0) begin //TNODEOUT = 0 starts here.
            if ($port_connected(p) && $port_connected(b)) begin
               V(b) <+ 0;
            end else begin
               if ($port_connected(p)) begin
                  V(p) <+ 0;
               end
            end
         end else begin //TNODEOUT = 0 for SHMOD = 0 ends here.
            if ($port_connected(p) && $port_connected(b) && $port_connected(t)) begin
               V(b) <+ 0; //Case 8
            end
         end         //TNODEOUT = 1 for SHMOD = 0 ends here.
      end else begin     //SHMOD = 0 ends here.
         if (TNODEOUT == 0) begin
            if ($port_connected(p) && $port_connected(b) && $port_connected(t)) begin
               V(b) <+ 0;
            end else begin
               if ($port_connected(p) && $port_connected(b)) begin
                  V(b) <+ 0;
               end else begin
                  if ($port_connected(p)) begin
                     V(p) <+ 0;
                  end
               end
            end
         end else begin //TNODEOUT = 0 for SHMOD = 1 ends here.
            if ($port_connected(p) && $port_connected(b) && $port_connected(t)) begin
               V(b) <+ 0;
```

```
                end
   end //TNODEOUT = 1 for SHMOD = 1 ends here.
  end //SHMOD = 1 ends here.
  // Following block is for issuing warnings for NODECHK
        if (NODECHK) begin      /* 2014bug2 */
            if (SHMOD == 0) begin
                if (TNODEOUT == 0) begin //TNODEOUT = 0 starts here.
                    if ($port_connected(p) && $port_connected(b)) begin
                        $strobe("Warning: 'P' and 'B' nodes were specified, however 'SOIMOD = 2' does not support 'P' or 'B' nodes. Internally these nodes are tied to global ground.");
                    end else begin
                        if ($port_connected(p)) begin
                            $strobe ("Warning: A 'P' node was specified, however 'SOIMOD = 2' does not support a 'P' node. Internally this node is tied to global ground.");
                        end
                    end
                end else begin //TNODEOUT = 0 for SHMOD = 0 ends here.
                    if ($port_connected(p) && $port_connected(b) && $port_connected(t)) begin
                        $strobe("Warning: 'P' and 'B' nodes were specified, however 'SOIMOD = 2' does not support 'P' or 'B' nodes. Internally these nodes are tied to global ground.");
                    end else begin
                        if ($port_connected(p) && $port_connected(b)) begin
                            $strobe ("Warning: A 'P' node was specified, however 'SOIMOD = 2' does not support a 'P' node. Internally this node is tied to global ground.");
                        end
                    end
                end  //TNODEOUT = 1 for SHMOD = 0 ends here.
            end else begin  //SHMOD = 0 ends here.
                if (TNODEOUT == 0) begin
                    if ($port_connected(p) && $port_connected(b) && $port_connected(t)) begin
                        $strobe("Warning: 'P' and 'B' nodes were specified, however 'SOIMOD = 2' does not support 'P' or 'B' nodes. Internally these nodes are tied to global ground.");
                    end else begin
                        if ($port_connected(p) && $port_connected(b)) begin
                            $strobe("Warning: 'P' and 'B' nodes were specified, however 'SOIMOD = 2' does not support 'P' or 'B' nodes. Internally these nodes are tied to global ground.");
                        end else begin
                            if ($port_connected(p)) begin
                                $strobe ("Warning: A 'P' node was specified however 'SOIMOD = 2' does not support a 'P' node. Internally this node is tied to global ground.");
                            end
                        end
                    end
                end else begin //TNODEOUT = 0 for SHMOD = 1 ends here.
                    if ($port_connected(p) && $port_connected(b) && $port_connected(t)) begin
                        $strobe("Warning: 'P' and 'B' nodes were specified, however 'SOIMOD = 2' does not support 'P' or 'B' nodes. Internally these nodes are tied to global ground.");
                    end else begin
                        if ($port_connected(p) && $port_connected(b)) begin
                            $strobe("Warning: Fifth and sixth nodes were specified when TNODEOUT = 1. The fifth node will be treated as the temperature node 'T', and internally the sixth node will be tied to global ground.");
                        end else begin
                            if ($port_connected(p)) begin
                                $strobe ("Warning: A fifth node has been specified with TNODEOUT = 1. The fifth node will be treated as the temperature node 'T'.");
                            end else begin
                                if (!($port_connected(p) &&
$port_connected(b) && $port_connected(t))) begin
                                                                                    $strobe("Warning: TNODEOUT = 1
requesting an external temperature node, but a fifth node for temperature is not supplied.");
                                                                            end
                                                                    end
                                                            end
                                                    end
                                            end //TNODEOUT = 1 for SHMOD = 1 ends here.
                                    end //SHMOD = 1 ends here.
                        end //NODECHK ends here.
```

16. 2016bug10:

a) In Spectre simulations, SHMOD value was not being updated as it was a model parameter, resulting in incorrect behavior.

Fix: Changed SHMOD to be an instance parameter

b) For the DGSEPB case (for SHMOD=1, TNODEOUT=1), Spectre was exiting with fatal error, complaining about shorted loops.

Fix: Omitted the redundant V(p) <+ 0 statement for the (d g s e p b) case as in this case we already have V(b, p) <+ 0 and V(b) <+0 which implicitly mean that the P node is shorted to ground too.

18. 2016enh2: Updated Copyright Statement

Copyright for ADI
 * Copyright (C) 2010, Analog Devices, Inc. All rights reserved.
 * The original Verilog-A implementation of BSIMSOI was developed by
 * Analog Devices, Inc.
 * Members of the Compact Model Council of the TechAmerica organization and
 * Tiburon Design Automation, Inc. are acknowledged for assistance.

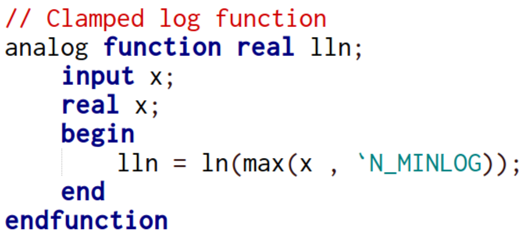
Copyright for UCB
* The original C source code is
* Copyright (C) 2016, Regents of the University of California. All rights reserved.

19. 2016enh3: Units and ranges

- Added units for most of the parameters
- Proper ranges were set for the parameters

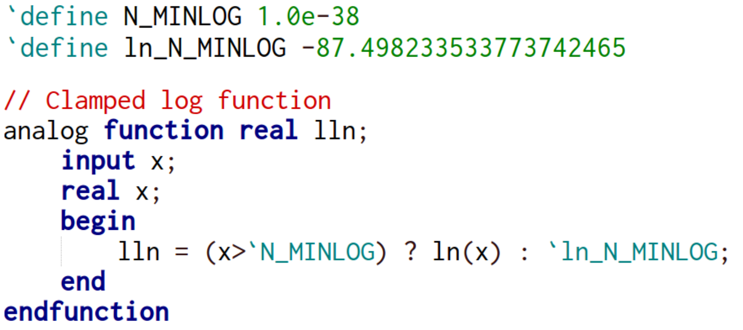
20. 2016enh4: Improved the clamped log function efficiency. Now **ln(`N_MINLOG)** is pre-evaluated so that it does not need to be recalculated each time **lln** is called.

Old Code



```
// Clamped log function
analog function real lln;
    input x;
    real x;
    begin
        lln = ln(max(x , `N_MINLOG));
    end
endfunction
```

New Code



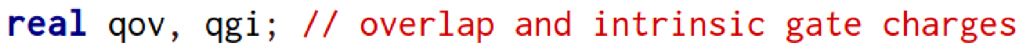
```
`define N_MINLOG 1.0e-38
`define ln_N_MINLOG -87.498233533773742465

// Clamped log function
analog function real lln;
    input x;
    real x;
    begin
        lln = (x>`N_MINLOG) ? ln(x) : `ln_N_MINLOG;
    end
endfunction
```

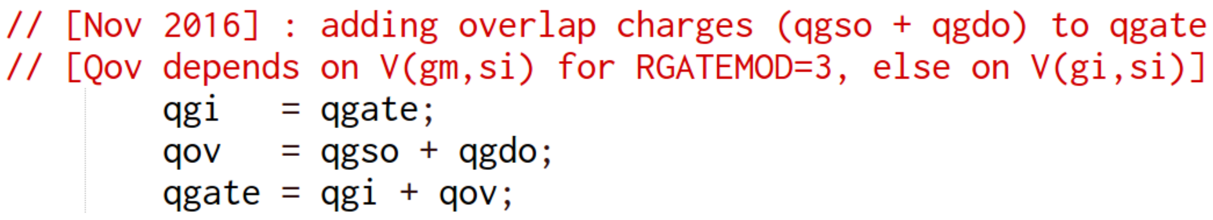
21. 2016enh5: Earlier QG didn't have any contribution from QOVERLAP.
Gate charge and gate capacitance calculations have now been modified to include overlap charge.

Modified Gate Charge Calculation

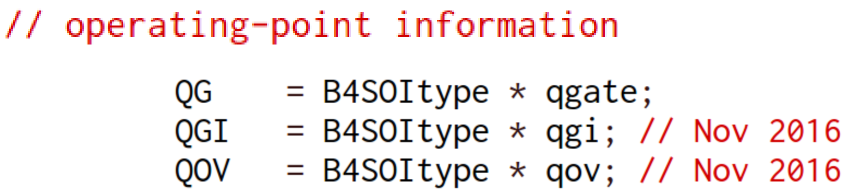
New variables:



```
real qov, qgi; // overlap and intrinsic gate charges
```



```
// [Nov 2016] : adding overlap charges (qgso + qdgo) to qgate
// [Qov depends on V(gm,si) for RGATEMOD=3, else on V(gi,si)]
        qgi   = qgate;
        qov   = qgso + qgdo;
        qgate = qgi + qov;
```



```
// operating-point information

        QG    = B4SOItype * qgate;
        QGI   = B4SOItype * qgi; // Nov 2016
        QOV   = B4SOItype * qov; // Nov 2016
```

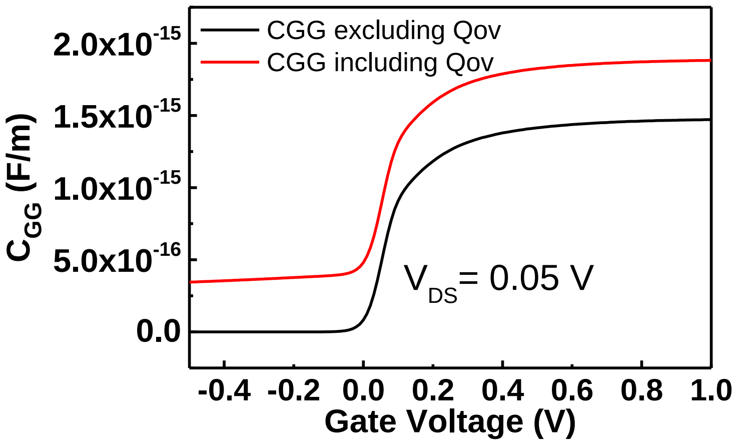
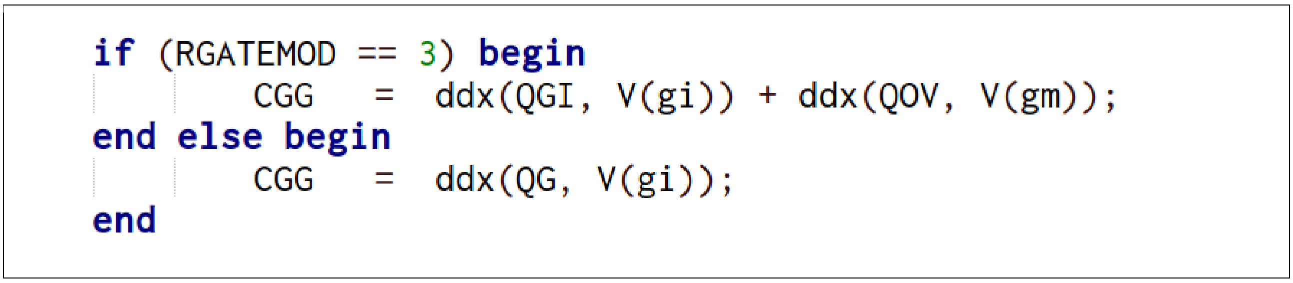


Figure 1: Comparison of CGG including and excluding overlap charge in the total gate charge

Modified CGG Calculation



```
if (RGATEMOD == 3) begin
        CGG    =  ddx(QGI, V(gi)) + ddx(QOV, V(gm));
end else begin
        CGG    =  ddx(QG, V(gi));
end
```

22. 2017enh1: Binning of Built-In Potential Lowering (ΔVbi) Model Parameters

- Binning of following parameters has been included.
    1. **Vbsa:** “vbsa offset voltage”
    2. **Vsce:** “SCE parameter for improved dVbi model”
    3. **Cdsbs:** “coupling from Vd to Vbs for improved dVbi model”
    4. **Vofffd:** “smoothing parameter in FD module”
    5. **Nofffd:** “smoothing parameter in FD module”
    6. **Moinfd:** “Coefficient for the gate-bias dependent surface potential in FD”
    7. **K1b:** “First backgate body effect parameter”
    8. **K2b:** “Second backgate body effect parameter”
    9. **Dk2b:** “Third backgate body effect parameter for short channel effect”
    10. **Dvbd0:** “First short-channel effect parameter in FD module”
    11. **Dvbd1:** “Second short-channel effect parameter in FD module”
    12. **Vbs0pd:** “Upper bound of built-in potential lowering for PD operation”
    13. **Vbs0fd:** “Lower bound of built-in potential lowering for PD operation”

23. 2017enh2: Output variables added for self-heating mode.

1. T_TOTAL_K : Output in Kelvin.
2. T_TOTAL_C : Output in Celsius.
3. T_DELTA_SH : Output for Self-Heating Temperature