BSIM3v3 Model Verilog-A Implementation

BSIM3v3 Model

Berkeley University BSIM3v3 model is the industry-standard, physics-based, deep-submicron MOSFET model for digital and analog circuit designs. This makes BSIM3v3 model a good candidate for implementation in Verilog-A HDL for study purpose or customized model use.

The SILVACO BSIM3v3 Verilog-A porting is based on the latest version BSIM3v3.2.4, released on December, 21st 2001. ***SmartSpice*** Verilog-A interface version 2.6.0.R has been used.

Verilog-A Porting

SILVACO BSIM3v3 Verilog-A implementation includes all the improvements of the original Berkeley model: ΔL and ΔW dependencies for different W and L devices, new capacitance model, new relaxation time model for characterizing the NQS effect, source/bulk and drain/bulk diode models. Additional model features such as Bulk-Drain and Bulk-Source GMIN, drain/ source inversion, N/P MOS type and parameter checking have also been added.

Implemented model selectors are:

```
    parameter integer MOBMOD = 1; //
      Mobility model selector
    parameter integer CAPMOD = 3; //
      Capacitance model selector
    parameter integer NQSMOD = 0; //
      Non-quasi-static model selector
```

The latest intrinsic capacitance model (Charge Thickness Model) is available with CAPMOD=3 and is set by default. Mobility model accounts for depletion mode devices (MOBMOD=2) and body bias dependence (MOBMOD=3). The NQS model subcircuit has been added and can be turned on by setting NQSMOD to 1 (NQS model is turned off by default).

L and W dependent parameter calculation as well as parameter checking are needed only once at the beginning of the simulation. Therefore, they have been implemented using `initial_step` event for simulation time saving:

```
@(initial_step)
  begin
  ...
  nsub  = nsub  + LNSUB  * Inv_L + WNSUB
  * Inv_W + PNSUB  * Inv_LW;
  ngate = ngate + lngate * Inv_L + wngate
  * Inv_W + pngate * Inv_LW;
  ...
  if (nsub <= 0.0)
     begin
        $strobe (“Fatal: Nsub = %g is
        not positive.”, nsub);
        Fatal_Flag = 1;
     end
  ...
end
```

DC equations are implemented in the analog part, for example Ids current:

$$Idso = \frac{\beta \cdot Vgsteff \cdot \left(1 - Abulk \cdot \dfrac{Vdseff}{2 \cdot (Vgsteff + 2 \cdot Vt)}\right) \cdot Vdseff}{1 + Vdseff/(Esat \cdot Leff)}$$

$$Ids = \frac{Idso}{1 + \dfrac{Rds \cdot Idso}{Vdseff}} \cdot \left(1 + \frac{Vds - Vdseff}{V_A}\right) \cdot \left(1 + \frac{Vds - Vdseff}{V_{ASCBE}}\right)$$

becomes in Verilog-A language:

```
analog begin
   ...
   fgche1 = Vgsteff * (1.0 - 0.5 * Abulk *
   dseff / Vgst2Vtm);
   fgche2 = 1.0 + (Vdseff / EsatL);

   gche = beta * fgche1 / fgche2;
   Idl  = gche * Vdseff / (1.0 + gche * Rds);

   Idsa = Idl  * (1.0 + (diffVds / Va));
   Ids  = Idsa * (1.0 + (diffVds / VASCBE));
   ...
   I(drain, source) <+ TYPE * Ids;
   ...
end
```

Once the Ids current is calculated, its contribution is added with <+ operator between the intrinsic drain and source nodes. Derivatives are automatically calculated so the written code is shorter than the compact model C-code and bugs in hand-written derivatives are no more possible. N/P MOS type contribution is accounted for with TYPE parameter.

The BSIM3v3 charge model is implemented the following way:

```
if (analysis(“tran”, “ac”))
   begin
     if (CAPMOD == 0)
       begin
         qgate  = ...;
         qdrain = ...;
       end
     else if (CAPMOD == 1)
       begin
         qgate  = ...;
         qdrain = ...;
       end
     else if (CAPMOD == 2)
       ...
   end
```

Intrinsic node charges are calculated according to `CAPMOD` model parameter. `analysis(“tran”, “ac”)` function call can be used to compute charges only in transient and ac analysis. Then, extrinsic elements such as overlap capacitances charges are calculated and added to the corresponding node, for example:

```
qgdo = Cgdo * Vgd;
Qgate  = qgate  + qgdo; // qgdo is
  gate/drain overlap capacitance
  charge
Qdrain = qdrain - qgdo;
```

The resulting charge current contribution is added to the circuit by using ddt time derivative operator:

```
I(gate)  <+ TYPE * ddt(Qgate); // Qgate
  is the total node charge
I(drain) <+ TYPE * ddt(Qdrain);
```

Validation

In transient analysis, a one-shot trigger test circuit supplied by Berkeley University team has been successfully run with BSIM3v3 Verilog-A model, with a reasonable simulation time. In AC analysis as well, an operational amplifier has been successfully simulated.

To improve the model convergence especially with large circuits, the MAXDELTA nature attribute is used to systematically limit the per-iteration voltage bias changes. GMIN conductances have been added to drain/bulk and source/bulk branches to improve convergence. GMIN is a model parameter and set to 1e-12 Ω^{-1} by default.

For Verilog-A implementation validation, results have been compared with ***SmartSpice*** internal BSIM3v3 model level=8.

Conclusion

Berkeley BSIM3v3.2.4 model has been successfully implemented in Verilog-A at SILVACO. The fact that the derivatives are automatically calculated reduces dramatically the implementation time and avoid introducing bugs in derivative calculation. It also increases the readability, shortening the 20,000 lines original Berkeley C-code to a 3,000 lines Verilog-A module. The Verilog-A simulation fit the results with ***SmartSpice*** internal model and shows a good convergence on large circuits with the help of voltage limitation. The Verilog-A model is freely available on SILVACO website **(http://www.silvaco.com).**

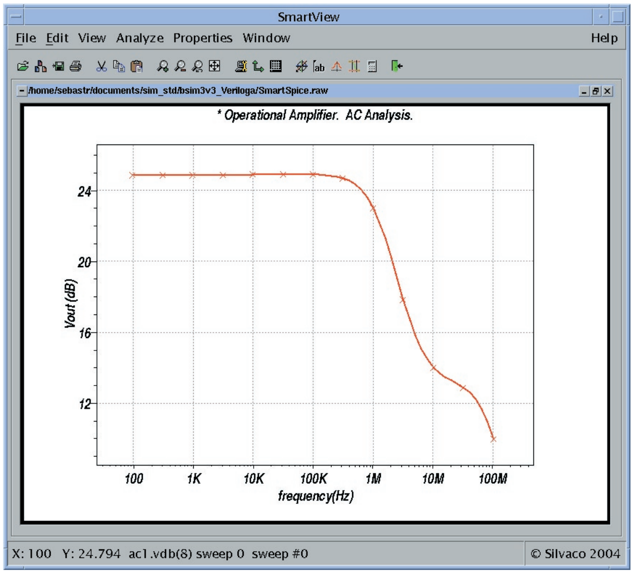

Figure 1. Operational amplifier AC output with BSIM3v3 Verilog-A model

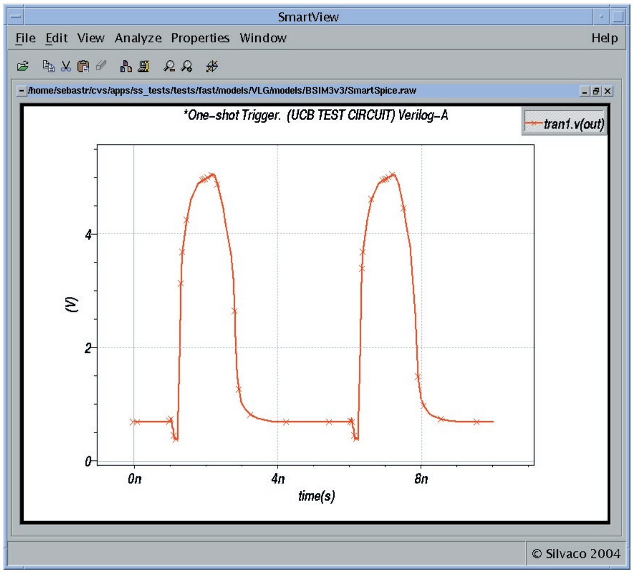

Figure 2. One-shot trigger output with BSIM3v3 Verilog-A model