Modeling Varactors

Ken Kundert
Designer's Guide Consulting, Inc.

Version 4, 19 June 2002

Describes the basic approach for developing a Verilog-A model of a varactor once a formula has been derived that gives its capacitance as a function of bias voltage.

Last updated on May 11, 2006. You can find the most recent version at www.designers-guide.org. Contact the author via e-mail at ken@designers-guide.com.

1 Capacitance

A capacitor is defined as being a component whose charge is a function of voltage. Its capacitance is defined as the derivative of charge with respect to voltage,

$$C(v) = \frac{dq(v)}{dv}. \tag{1}$$

The current through a capacitor is simply the time-derivative of the charge,

$$i(t) = \frac{dq(v(t))}{dt}. \tag{2}$$

This can be expanded to

$$i(t) = \frac{dq(v(t))}{dv(t)}\frac{dv(t)}{dt} = C(v(t))\frac{dv(t)}{dt}. \tag{3}$$

2 Charge Conservation

Though (2) and (3) are equivalent, using (3) to build models for use in a simulator is problematic. Simulators break time into discrete steps and solve the circuit equations at the points between the steps. The same $C(v)$ is used across each step, which results in small errors on every step, which further leads to the charge not being conserved if the capacitor is nonlinear (see section 4.2.5, pg. 167 of [1]). A small amount of charge is either created or annihilated on each step. The amount of charge that is not conserved is small if tolerances are tight. However, the problem can be quite significant in many typical cases. A tell-tale symptom of this problem is that an anomalous DC current appears to flow through the varactor. This problem does not occur if (2) is used.

To understand why nonlinear capacitor models implemented using (3) do not conserve charge, consider a capacitor with capacitance

$$C(v) = C_0 + C_1 v. \tag{4}$$

Assume the voltage across the capacitor changes from $v_0 = 0$ to $v_1 = 1$ and then back to $v_2 = 0$ in two steps. The change in charge is then computed by applying a backward Euler approximation [1] to (3),

$$\Delta q_k = C(v_k)\Delta v_k. \tag{5}$$

On the first step,

$$\Delta q_1 = C(v_1)\Delta v_1 = C(1)1 = C_0 + C_1. \tag{6}$$

On the second step,

$$\Delta q_2 = C(v_2)\Delta v_1 = C(0)(-1) = -C_0. \tag{7}$$

Since $v_0 = v_2$, Δq_1 and Δq_2 must sum to zero for charge to be conserved,

$$\Delta q_1 + \Delta q_2 = C_0 + C_1 - C_0 = C_1. \tag{8}$$

Thus, charge is only conserved if $C_1 = 0$, which implies that the capacitor must be linear to conserve charge when using (3).

Conversely, for capacitor models implemented using (2), the change in charge over a step becomes

$$\Delta q_k = q(v_k) - q(v_{k-1}), \tag{9}$$

where

$$q(v) = \int (C_0 + C_1 v) dv = C_0 v + \frac{C_1 v^2}{2}. \tag{10}$$

As before, with $v_0 = v_2$ Δq_1 and Δq_2 must sum to zero for charge to be conserved,

$$\Delta q_1 + \Delta q_2 = q(v_1) - q(v_0) + q(v_2) - q(v_1) = -q(v_0) + q(v_2) = 0. \tag{11}$$

and so charge is always conserved when using (2).

The essential point to take away is that with models formulated using (3) the charge must be estimated from the capacitance. There are different ways of doing the estimation. The backward Euler approximation given above estimates the change in charge from the capacitance at the end of the step; forward Euler uses the capacitance at the beginning of the step, and trapezoidal rule uses a combination. Each produces an error that grows with the time step that causes charge to not be conserved. With models formulated using (2) the charge is computed explicitly, so there is no approximation.

In addition to the charge conservation problems associated with (3), Spectre's implementation of Verilog-A complains about a "linear accessibility" problem if you construct a model directly from (3) and C is nonlinear. It is possible to overcome this problem by using a node rather than a variable to hold *dv/dt*. However, this can cause scaling problems if you are not careful, and is computationally more expensive to simulate. These issues, combined with the charge conservation problem, make it unappealing to base a model on (3).

3 A Common Modeling Error

A common mistake is to formulate the model for a nonlinear capacitor by starting with the model of a linear capacitor. If a capacitor is linear its charge is

$$q(v) = Cv, \tag{12}$$

and so the current through the capacitor is

$$i(t) = \frac{d(Cv(t))}{dt}, \tag{13}$$

$$i(t) = C\frac{dv(t)}{dt}. \tag{14}$$

Modeling a nonlinear capacitor by replacing C with $C(v)$ in (14) is identical to using (3), which as mentioned above, does not conserve charge. However, the situation is even worse if C is replaced with $C(v)$ in (13). It is important to recognize that

$$\frac{d(C(v(t))v(t))}{dt} \neq C(v(t))\frac{dv(t)}{dt} \tag{15}$$

because $C(v(t))$ itself varies with time. Thus, using

$$i(t) = \frac{d(C(v(t))v(t))}{dt} \tag{16}$$

produces large errors if C is a strong function of v and v varies significantly with t.

These types of problems are common whenever one tries to model any nonlinear component by simply specifying the incremental parameter (C, L, R, etc.) as a nonlinear function to an otherwise linear model. This occurs with simulators such as HSPICE and Eldo when you specify the R, C, or L of the built-in resistor, capacitor, or inductor models as being a function of either voltage or current. In this case, the resistance, capacitance, or inductance actually exhibited by the component is *guaranteed* to be different from what you specified, often substantially so (see Figure 2 for an example).

4 Deriving a Charge-Based Model

When $C(v)$ is known, the proper approach to building a large signal model is to reconstruct q as a function of v. Recall that capacitance is the derivative of charge with respect to voltage. Thus, the charge q is simply the integral of the capacitance C with respect to the voltage v.

$$q(v) = \int_0^v C(v)dv \tag{17}$$

Then, the current is computed using

$$i(t) = \frac{dq(v(t))}{dt} \tag{18}$$

Using (18) to construct a model results in a model that is both accurate and computationally efficient. It suffers neither the charge conservation problems common to (3) and (16), nor from the accuracy problems of (16).

5 An Example

Consider a varactor (a nonlinear capacitor) with the following characteristics

$$C(v) = C_0 + C_1 \tanh\left(\frac{v - v_0}{v_1}\right) \tag{19}$$

The capacitance for this model is shown in Figure 1.

The original model for this varactor was given as

$$i = \frac{d(C(v)v)}{dt} \tag{20}$$

Since the time derivative was applied to the product of $C(v)v$ and not just to v, the mistake of (16) was made. The capacitance of the resulting model is shown in Figure 2. The capacitance exhibited by this model differs substantially from what is expected. For the choice of parameters used in this example (C_0 = 1 pF, C_1 = 0.5 pF), the error is more than 30%. However, if C_1 is increased to 0.85 pF, the effective capacitance becomes

FIGURE 1 *Capacitance of the varactor as a function of voltage.*

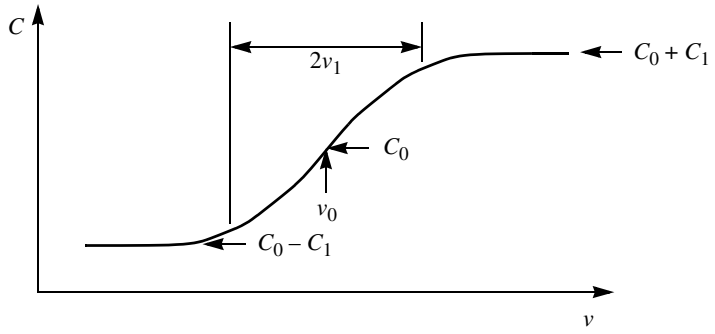
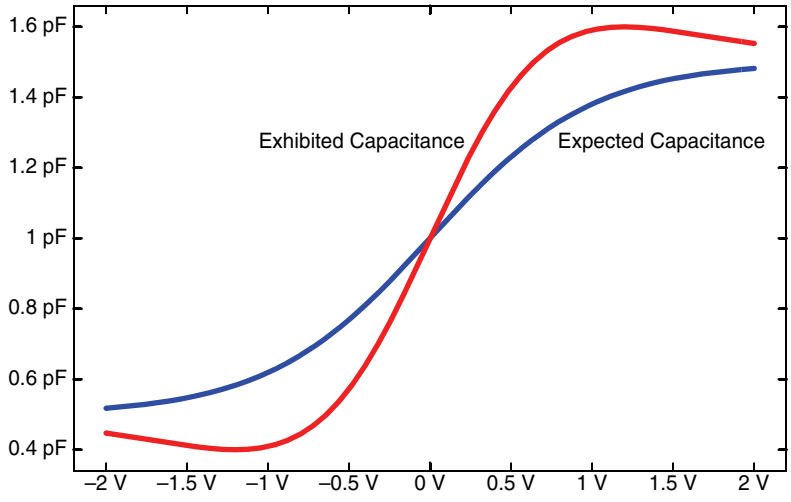


FIGURE 2 *The capacitance exhibited by a model based on (20) is substantially different from what is expected.*



negative over part of its range, which results in otherwise stable circuits appearing to be unstable.

The model is properly derived by applying (17) to (19).

$$q(v) = \int_0^v \left(C_0 + C_1 \tanh\left(\frac{v - v_0}{v_1}\right)\right) dv = C_0 v + C_1 v_1 \ln\cosh\left(\frac{v - v_0}{v_1}\right) \tag{21}$$

and then, of course,

$$i(t) = \frac{d(q(v(t)))}{dt} \tag{22}$$

6 Example as a Verilog-A Model

This model is formulated in Verilog-A [2,3] as shown in Listing 1. It is a direct translation of (21) and (22). Limits were added on *c0*, *c1*, and *v1* to assure the capacitance

exhibited by the varactor is positive. If the capacitance were allowed to become negative, the circuit would likely be unstable, which besides causing the results to be wrong, also could cause serious numerical problems.

LISTING 1 *Varactor model.*

```
`include "discipline.vams"

module varactor(p, n);
inout p, n;
electrical p, n;
parameter real c0 = 1p from (0:inf);      // nominal capacitance (F)
parameter real c1 = 0.5p from [0:c0);     // maximum capacitance change from nominal (F)
parameter real v0 = 0;                    // voltage for nominal capacitance (V)
parameter real v1 = 1 from (0:inf);       // voltage change for maximum capacitance (V)
real q, v;

    analog begin
        v = V(p,n);
        q = c0*v + c1*v1*ln(cosh((v – v0)/v1));
        I(p, n) <+ ddt(q);
    end
endmodule
```

7 Verifying the Model

You can easily plot the capacitance of the varactor by driving the capacitor with a voltage source with unity AC magnitude [1]. Perform an AC analysis at $1/2\pi$ Hertz while sweeping the DC source voltage over the desired range of values. This can be done while running Spectre standalone with a netlist like the one shown in Listing 2.

LISTING 2 *Test circuit.*

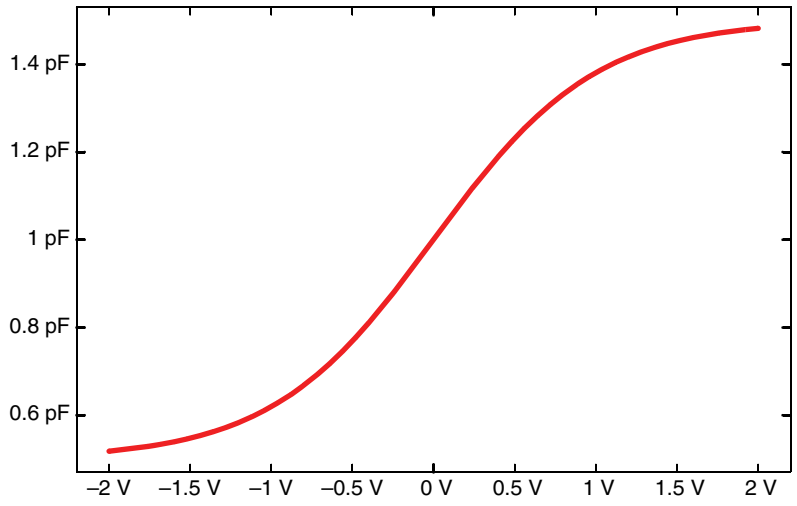
```
// varactor test circuit
simulator lang=spectre
ahdl_include "varactor.va"

Vt (n 0) vsource mag=1
Cv (n 0) varactor c0=1pF c1=0.5pF v0=0 v1=1

capacitanceInF ac freq=1/(2*M_PI) start=–2 stop=2 dev=Vt param=dc
save Cv:1
```

The output is shown in Figure 3. The graph was created by plotting the current through the capacitor and changing the units from Amperes to Farads.

8 If You Have Questions

If you have questions about what you have just read, feel free to post them on the *Forum* section of *The Designer's Guide Community* website. Do so by going to *www.designers-guide.org/Forum*.

FIGURE 3 *Capacitance as a function of voltage measured using simulation.*



References

[1] Kenneth S. Kundert. *The Designer's Guide to SPICE and Spectre*. Kluwer Academic Publishers, 1995.

[2] Kenneth S. Kundert. *The Designer's Guide to Verilog-AMS*. Kluwer Academic Publishers, 2004.

[3] *Verilog-AMS Language Reference Manual: Analog & Mixed-Signal Extensions to Verilog HDL*, version 2.1. Accellera, January 20, 2003. Available from *www.accellera.com*. An abridged version is available from *www.verilog-ams.com* or *www.designers-guide.org/VerilogAMS*.