A Generic Formalism to Model ESD Snapback for Robust Circuit Simulation

Tianshi Wang [†] and Colin C. McAndrew [‡]
[†] The Department of Electrical Engineering and Computer Sciences, University of California, Berkeley
Address: 545J Cory Hall, UC Berkeley, CA, USA, 94720
Email: tianshi@berkeley.edu
Phone: (510) 612-2379
[‡] NXP Semiconductors, AZ, USA

Abstract—**This paper introduces a way of modeling the abrupt turn-on/off behavior of ESD protection devices using entirely continuous and smooth equations. It presents accurate and robust ESD snapback models that are convenient and flexible to use for various types of ESD protection devices without convergence issues during simulation.**

I. Introduction

ESD protection devices commonly operate based on a mechanism known as snapback — the current through such a device does not grow monotonically with the voltage, but folds back in a certain voltage range. This phenomenon often causes convergence difficulties during circuit-level simulation, calling for the development of more robust compact models for such devices.

To overcome these difficulties, early models were designed to work with dedicated algorithms [1–3] that are no longer available in today's open-source or commercial simulators. Later modeling efforts aim to combine existing compact models for semiconductor devices (MOSFETs, BJTs, *etc.*) and represent ESD clamps using equivalent circuits [4–9]. They commonly require the use of if-else conditions and external current sources; these non-physical constructs often create new convergence issues. Also, these models are normally computationally expensive, and have many more parameters than needed for capturing the clamp's operation, making parameter extraction from measurements difficult. Because of their complexity, debugging is also difficult when there are issues with either accuracy or convergence. In comparison, behavioral compact models [10–12] with simple equations and easy-to-characterize parameters are becoming more and more attractive for practical application. However, existing behavioral models still rely on if-else discontinuities to work around the abrupt switching between on and off states. Some also use Boolean or integer variables [10, 11] to implement state-machine-style transitions. Such a practice does not comply with compact modeling guidelines [13, 14] and can often compromise the robustness of the models — the resulting "hybrid" models are not compatible with the way analog simulators are written, and often produce inconsistent results in different simulators.

In this paper, we first discuss the generic structure for the behavioral models of ESD protection devices in Sec. II. The structure consists of two equations — one specifies the *I–V* relationship, the other describes the dynamics of an internal unknown variable representing the occurrence of impact ionization. By designing a fold in the DC solution curve of the second equation, snapback in the *I–V* graph and the abrupt turn-on/off behavior come about naturally from these entirely continuous and smooth model equations. Then we apply this formulation to ESD protection devices, introducing a model that captures the snapback mechanism accurately, has parameters easily extractable from measurement data, and works reliably in various circuits. We then present the Verilog-A implementation of this model in Sec. III, focusing on using the Verilog-A language to write internal unknown variables and implicit differential equations in a way that is compatible with all analog circuit simulators, such that the model generates consistent results in them. Simulation examples and the parameter extraction procedures are discussed in Sec. IV.

The model template we discuss in this paper is not limited to using the provided fitting functions; it can be augmented with more complex physics-based equations for fitting measurements from various types of ESD protection devices. As examples, we extend the proposed model to incorporate the second snapback phenomenon in Sec. V, and also apply it to model snapback in multi-terminal devices in Sec. VI.

II. Compact Modeling of ESD Snapback

A general two-terminal device, such as an ESD clamp in its simplest form, has a branch voltage V and a branch current I; its behavioral compact model should provide equations describing the relationship between them.

Normally, we would like to express I as a function of V:

$$I = \frac{d}{dt}q(V) + f(V). \tag{1}$$

However, for ESD clamps, the current is not determined by the instantaneous voltage alone; it also depends on the device's on/off state. So we rewrite (1) as

$$I = \frac{d}{dt}q(V) + f_1(V,\ s), \tag{2}$$

where s is an internal state variable representing the occurrence of impact ionization; its dynamics can be modeled with the help of an extra differential equation.

$$\frac{d}{dt}s = f_2(V,\ s). \tag{3}$$

Equations (2) and (3) constitute a complete behavioral model template for a general two-terminal clamp; the template has one internal unknown and one implicit differential equation so as to make modeling snapback possible. For the model to be robust in simulation, all functions — $q()$, $f_1()$ and $f_2()$ should be continuous and smooth. In the remainder of this section, we show that carefully designing these smooth functions can bring out the snapback behavior naturally.

To model the I–V relationship of ESD clamps using equation (2), we can separate the current I into two parts: I_{off} and I_{on}. The off state current I_{off} is often written as the junction leakage current [12][1]:

$$I_{off}(V) = I_S \cdot e^{-V/V_T} \cdot \sqrt{1 + \frac{\max(V,\ 0)}{V_D}}, \tag{4}$$

where I_S is the reverse saturation current of P-N junctions, V_T is the thermal voltage, V_D is the critical voltage of P-N junctions.

I_{on} represents the extra current flowing through the device in the on state:

$$I_{on}(V) = I_{h1} + G_{on1} \cdot (V - V_{h1}), \tag{5}$$

where V_{h1} is the holding voltage, I_{h1} is the holding current, G_{on1} is the on-state conductance — all these parameters can be easily estimated from measurements.

If we let $s = 0$ represent the device being entirely off, and $s = 1$ represent entirely on, we can put together a formula for the total current I as[2]

$$I = \frac{d}{dt}q(V) + f_1(V,\ s) = \frac{d}{dt}C \cdot V + I_{off}(V) + s \cdot I_{on}(V), \tag{6}$$

where in addition to the I_{off} and I_{on} discussed above, we use a linear capacitor as the charge model as we focus mainly on the modeling of I–V snapback in this paper. More accurate charge models [12] can be used in this formulation without modifications to the equation structure.

The internal state variable s in (6) represents the occurrence of impact ionization. It should start to grow to 1 when V increases past the trigger voltage V_{t1}, and begin to decrease towards 0 when V drops below V_{h1}. As mentioned in Sec. I, existing behavioral models attempt to reproduce this phenomenon through the use of if-else statements, together with the use of Boolean or integer states to record the transitions, resulting in instantaneous turn-on/off behaviors. A recently developed model [12] uses an RC circuit to smoothly switch s on and off, but the two thresholds V_{t1} and V_{h1} are still set up using if-else discontinuities — it still does not comply with the formalism presented in (2) and (3). In this paper, we identify that the key to modeling the turn-on/off behaviors physically is to design the $f_2()$ function with a fold in its DC solutions. Suitable bivariate functions with this property are not hard to find; one example is provided below.

$$\frac{d}{dt}s = f_2(V,\ s) = \tanh(K \cdot (V + s)) - s. \tag{7}$$

From Figure 1 (a), the curve of $f_2(V,\ s) = 0$ in the $(V,\ s)$ plane folds back in the middle, creating multiple stable $f_2(V,\ s) = 0$ solutions when V is between $V-$ and $V+$. When $V < V-$, there is only one point with $s \approx -1$, satisfying $f_2(V,\ s) = 0$, representing the only steady state solution at this low biasing voltage. As V increases, s stays close to -1 until V reaches $V+$. Beyond $V+$, there is only one $f_2(V,\ s) = 0$ solution with $s \approx 1$, indicating that as V slowly increases past $V+$, s will suddenly jump to 1. Similarly, when voltage decreases, s remains at 1 until V passes $V-$, then s starts to transit to -1. The negative-sloped fold in the middle of the $f_2(V,\ s) = 0$ curve forms naturally as f_2 is continuous and smooth; combined with $f_1()$, the fold in the s–V plane creates the snapback curve in the I–V graph.

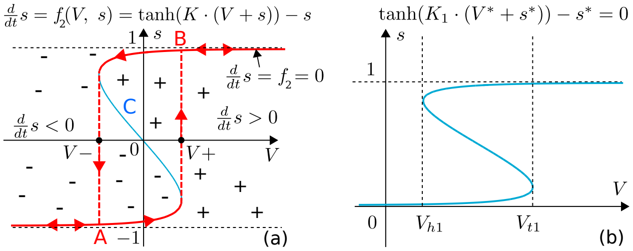

Fig. 1: (a) Signs and zero-crossings of an example $f_2()$ function, with $K = 2$; (b) zero-crossings of $f_2()$ after shifting in V–s plane.

It is worth highlighting that no if-else conditions are required for generating the switching behaviors described above; the turn-on/off dynamics come about from entirely

[1] In the model implementation, we also consider the soft breakdown before snapback. This is detailed in Sec. IV and can also be seen in the model code presented in Listing 1.

[2] Note that in (6), the portion of I_{on} we add to I is linear with respect to s, which is an arbitrary choice. Using nonlinear mapping functions instead can provide more flexibility for fitting measurements. We discuss this in more detail in Sec. IV when introducing model parameters.

smooth model equations thanks to the negative-sloped fold designed in the solutions of the $f_2(V,\ s)$ function. It is also worth noting the use of smooth model equations does not promise convergence in some simulation analyses, especially the simple Newton-Raphson-based DC analysis. In fact, even in the above discussion, we have mentioned the abrupt transitions at $V+$ and $V-$ that are unavoidable when the conventional DC sweep method forces the voltage to step in a fixed direction. However, this is more a limitation of the simulation algorithm than a flaw in the model. More advanced analyses, such as homotopy/continuation methods [15], can guarantee global DC convergence on circuits consisting of such smooth models.

To make the above model template suitable for ESD protection devices, we can shift the curve in Figure 1 (a) using the following transformations:

$$f_2(V,\ s) = \tanh(K \cdot (V^* + s^*)) - s^*, \tag{8}$$

where

$$\begin{aligned} V^* &= \frac{2}{V_{t1} - V_{h1}} \left(V - 0.5V_{t1} - 0.5V_{h1}\right), & (9) \\ s^* &= 2s - 1. & (10) \end{aligned}$$

V^* is designed to shift transition points in (7) from approximately -1 and 1 to V_{h1} and V_{t1}.[3] s^* aims to change the range of internal state variable in (7) from $s \in (-1, 1)$ to $s \in (0, 1)$. In this way, the device will turn on and off at the desired locations, with the internal state variable s shifting between 0 and 1.

Equations (6) and (8) constitute a behavioural model for ESD protection devices.

III. Model Implementation in Verilog-A

The Verilog-A language [16] models a device using a circuit structure, *i.e.*, with internal nodes and branches defined, similar to a SPICE subcircuit. The variables in a Verilog-A model, the "sources" and "probes", are potentials and flows specified based on this circuit topology. Coming from this subcircuit perspective, the language does not provide a straightforward way of dealing with general internal unknowns and implicit equations inside the model, *e.g.*, the state variable s and the equation (8).

As a result, there can be many pitfalls when one tries to write a behavioral ESD snapback model in Verilog-A. The internal unknown is often declared as a general variable using the `real` statement. `idt()`, `$abstime` and hard-coded time integration methods are often used for describing implicit differential equations. These approaches should be avoided in modeling [13]. Instead, in our implementation, we model the state variable s by considering it as a voltage, and code the implicit equation by treating it as the KCL at an internal node. As shown in the Verilog-A code in Listing 1, we declare an internal branch, whose voltage represents s. At one end of the branch is an internal node that does not connect to any other branches. In this way, by contributing $\tanh(K \cdot (V^* + s^*)) - s^*$ and `ddt(-tau * s)` both to this same branch, the KCL at the internal node will enforce that the implicit differential equation in (8) is always satisfied in any circuit analysis.

Declaring s as a voltage is not the only way to model internal unknowns in Verilog-A. Depending on the physical nature of s, one can also use Verilog-A's multiphysics support and model it as a potential in other disciplines. One can also switch potential and flow by defining s as a flow instead. The essence of our approach is to force all simulators to recognize the state variable s as a circuit unknown by modeling it as a potential or flow in Verilog-A, such that the model can be supported consistently by different simulators in various circuit analyses. For instance, the Verilog-A code in Listing 1 has been tested to generate consistent results in several simulation platforms, including Spectre®,[4] HSPICE,[5] and the open-source simulator Xyce.[6]

IV. Simulation Examples and Parameter Extraction

The resulting model can capture the *I–V* characteristics of ESD protection devices accurately. As an example, Figure 2 shows the operating points of the Verilog-A model with its default parameters, which are chosen based on the TLP measurement data from a Zener-breakdown-triggered bipolar clamp in a 0.25μm BiCMOS process [17]. By overlaying the simulated curve with measurements, we see that the model reproduces the *I–V* characteristics well, not only in the on and off states, but also in the snapback region.

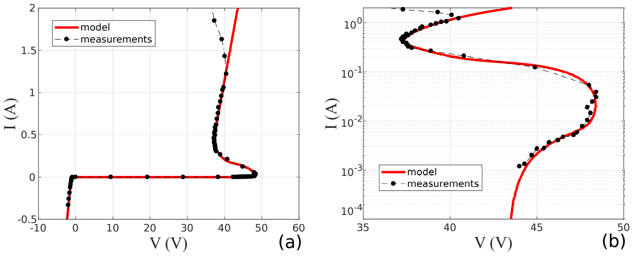

Fig. 2: TLP measurements *vs.*model's operating points, in linear and log scales.

While Figure 2 shows the quasi steady states of an ESD protection device under TLP testing, transient responses

[3] These transition points are only accurate when K is large. The inaccuracy can be compensated by scaling V^* by a correcting factor of $\sqrt{1-1/K} - 1/K \cdot \mathrm{arctanh}(\sqrt{1-1/K})$, as shown in the attached Verilog-A model code in Listing 1.

[4] Spectre® version: 15.1.0 64bit.

[5] HSPICE version: J-2014.09 64bit.

[6] Xyce version: 6.5.

of TLP can further reveal the time-domain information about the device's turn-on behavior. We have simulated the model's transient responses under pulses of different heights in a 50Ω TLP testing environment, and compared the results with measurements. From Figure 3 and Figure 4, we observe that even with a simple linear charge model and a single fixed parameter τ governing the turn-on time constant, the simulation results from the model are in good agreement with measurements, especially in the voltage overshoot magnitude and current settling speed.

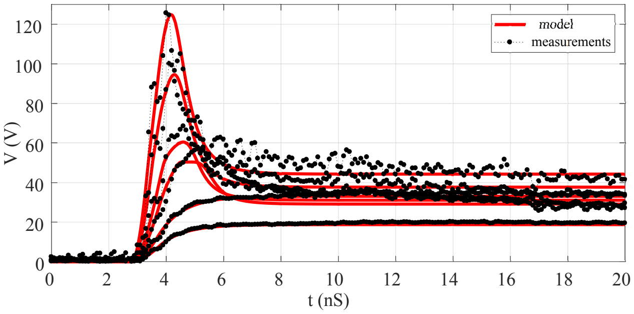

Fig. 3: Simulated *vs.*measured transient voltage responses under pulses with different heights.

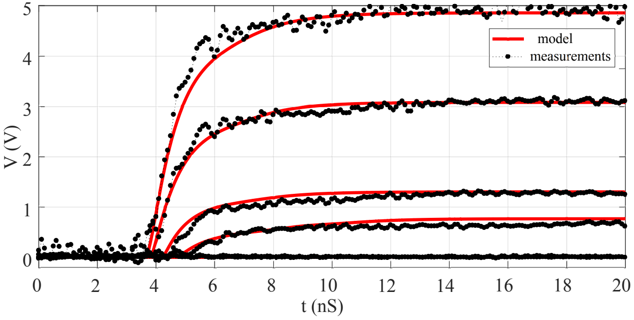

Fig. 4: Simulated *vs.*measured transient current responses under pulses with different heights.

To further test the model's robustness, we simulate it in the equivalent circuits of several ESD tests, constructed by changing the values of L1, C1 and R1 in the test bench circuit shown in Figure 5 (a). The three common ESD tests — the human-body model (HBM), the machine model (MM), and the charged-device model (CDM) — expose the ESD clamp to various environments and are useful for testing the model under different operating conditions. Simulation results of the voltage across the clamp and the current through the source are shown in Figure 5. In all these scenarios, we have not experienced any convergence issues, and the simulations produce the desired results.

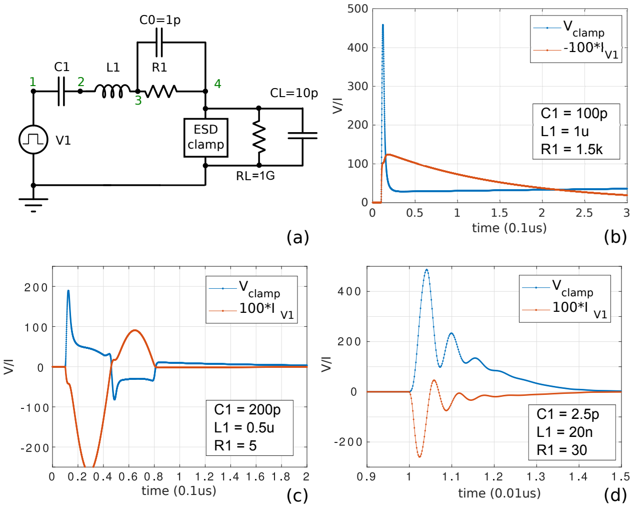

Fig. 5: (a) Equivalent circuit of ESD tests; (b-d) simulation results from HBM, MM and CDM tests.

One key feature of the proposed behavioral model is that it does not have many parameters, and its parameters can be organized into several groups with each group independently characterizing a single trait of the device. This feature makes parameter extraction from measurement data straightforward. Descriptions of the complete set of parameters are listed in Table I. Here we describe the organization of parameters as well as their extraction procedures as follows.

- `Vt1, It1, Vh1, Ih1`: parameters for the trigger and holding voltage/current; they can be conveniently estimated from the position of the snapback region in the *I*–*V* graph.
- `Gon1`: the on-state conductance; it can be determined through line fitting of the on-state *I*–*V* curve.
- `Is, VD, Gmax`: parameters related to the off-state current. As the off-state current is essentially junction leakage, these are standard parameters normally determined by fitting experimental data, or calculated directly based on junction doping. Among them, `Gmax` is the maximum conductance of the device's off state, beyond which the exponentially growing junction current is clipped to be linear with respect to the voltage increase. It not only physically captures the parasitic resistance present in the device, but also improves the numerical robustness of the model.
- `Vt0, Vsbd, smoothing_sbd`: parameters for characterizing the soft breakdown current, with the following formula [18]:

$$I_{sbd} = K_i \cdot \exp\left(-\frac{V_{sbd}}{V - V_{t0}}\right), \qquad (11)$$

where V_{t0} is the voltage where soft breakdown begins to occur; V_{sbd} is a fitting parameter governing the growth rate of the breakdown current with respect to voltage. Because soft breakdown provides the current I_{t1} at the trigger voltage V_{t1}, scaling factor K_i can be calculated from model parameters:

$$K_i = I_{t1} \Big/ \left(\exp\left(-\frac{V_{sbd}}{V - V_{t0}}\right) \cdot (V_{t1} - V_{t0})\right). \qquad (12)$$

Name	Default	Unit	Description
`Vt1`	50	V	trigger voltage for the first snapback
`It1`	1e-3	A	trigger current for the first snapback
`Vh1`	30	V	holding voltage for first snapback
`Ih1`	1	A	holding current for first snapback
`Gon1`	0.14	S	"on" state conductance after first snapback
`C`	1e-12	F	parallel capacitance
`tau`	1e-8	s	time constant of impact ionization
`Is`	1e-12	A	reverse saturation current of PN junction
`VT`	0.026	V	thermal voltage, can be replaced by $vt
`VD`	1	V	DC leakage early voltage
`Gmax`	2	S	maximum conductance under reverse bias
`K1`	10		fitting parameter for adjusting the "sharpness" of snapback: the larger, the sharper, the closer s is to {0,1}
`Alpha1`	5		fitting parameter for adjusting the shape of snapback: the larger, the rounder the snapback around `Vh1`
`Beta1`	0.1		fitting parameter for adjusting the shape of snapback: the larger, the rounder the snapback around `Vt1`
`smoothing`	1e-10		smoothing factor, normally very small
`Vt0`	45	V	trigger voltage for soft breakdown before snapback
`Vsbd`	10	V	tuning parameter for soft breakdown before snapback
`smoothing_sbd`	1e-3		smoothing factor for soft breakdown before snapback

TABLE I: ESD Snapback Model Parameters

During parameter extraction, `Vt0` can be directly read out from the *I–V* measurements, while `Vsbd` can be calculated through a line search minimizing the error between formula (11) and measurements in the soft breakdown region.

As equation (11) is only valid within the soft breakdown region, smooth functions are used to add I_{sbd} to the off-state current only within $(V_{t0},\ V_{t1})$ while keeping the region $(-\infty,\ V_{t0})$ unaltered. The switching at V_{t0} is controlled by a smoothing factor `smoothing_sbd`. In practice, we find that as long as `smoothing_sbd` is small enough, it does not change the accuracy of the model; it is usually left at its default value.

- `C`: the charge model is independent from the other parameters discussed so far, and the capacitance `C` can often be directly measured. Bias-dependent capacitance models [12] can also be incorporated into our formalism with the help of more model parameters.
- `tau`: determining the time constant of impact ionization requires measuring the transient responses of ESD clamps. Once the quasi-steady-state *I–V* curve has been fitted and the other parameters determined, a line search can be performed to calculate this parameter by minimizing the error from transient measurements.
- `Alpha1, Beta1`. As mentioned in Sec. II, we have chosen $f_2()$ based on a simple $\tanh()$ function to generate the negative-sloped fold required by the formalism. To have more control over the fold in the *I–V* graph, we use two fitting parameters, α and β to modulate the shape of the snapback curve. Instead of adding the on-state current linearly with s, we change the on-current term in (6) to have a nonlinear mapping function:

$$s \cdot I_{on} \longrightarrow g(s) \cdot I_{on} = (s^{\alpha} + s\beta)/(1+\beta) \cdot I_{on}. \quad (13)$$

The mapping does not alter the current in either the on or off state, as $g(1) = 1$ and $g(0) = 0$, but the use of α and β changes the shape of the snapback curve in the middle. Specifically, as illustrated in Figure 6, α and β adjust the "sharpness" of the turning points at V_{h1} and V_{t1} respectively.

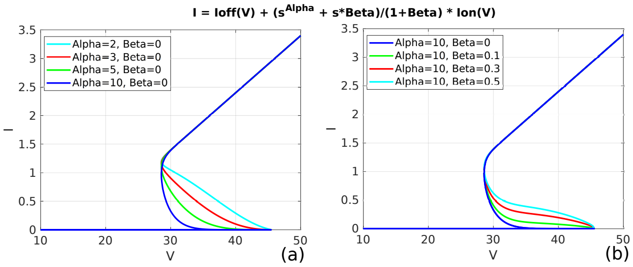

Fig. 6: *I–V* characteristic curves with different sets of α and β values.

The selection of these fitting parameters has also been automated. As TLP testing normally generates a lot more data near the $(V_{h1},\ I_{h1})$ corner, α is first extracted based on the corner's curvature; then with a fixed α, β is determined by minimizing the error from the rest of the measurements in the snapback region.

- `K1, smoothing`. `K1` modulates the smoothness around the $(V_{t1},\ I_{t1})$ and $(V_{h1},\ I_{h1})$ corners by changing the sharpness of the tanh() function in (8). When it is large enough, it creates virtually no change to the model. Similarly, `smoothing` is used in the smooth functions in the model and needs to be sufficiently small. In practice, these two parameters are usually left at their default values.

  It is worth noting that although these parameters do not have much effect on the model's accuracy, they can be used to further improve the model's convergence. As these parameters modulate the model's smoothness, sweeping them towards their desired values — analogous to GMIN stepping in SPICE — provides another option for aiding convergence during circuit simulation.

V. Incorporating Second Snapback

The formulation we introduce in this paper is very general — it can be easily adapted to incorporate the second snapback phenomenon, simply by adding another internal state variable. Suppose we have a variable s_2 that represents the occurrence of second snapback; it turns on and off at V_{t2} and V_{h2} respectively. We can write a differential equation similar to (8) to describe its dynamics:

$$\frac{d}{dt}s_2 = \text{smoothstep}(s^*)\cdot(\tanh(K_2\cdot(V_2^*+s_2^*))-s_2^*), \quad (14)$$

where

$$\begin{aligned} V_2^* &= \frac{2}{V_{t2}-V_{h2}}(V-0.5V_{t2}-0.5V_{h2}), && (15)\\ s_2^* &= 2s_2-1, && (16)\end{aligned}$$

such that s_2 turns on and off at the proper thresholds, but the dynamics is only in effect when $s^*>0$, *i.e.*, the device is already in the on state for the first snapback.

Then this second internal unknown variable s_2 modulates the contribution of a second on-state current to the total current, changing $f_1()$ in (6) to the following formula.

$$f_1(V,\ s) = I_{off}(V)+g(s)\cdot I_{on}(V)+g_2(s_2)\cdot I_{on2}(V), \quad (17)$$

where

$$\begin{aligned} g_2(s_2) &= (s_2^{\alpha_2}+s_2\cdot\beta_2)/(1+\beta_2), && (18)\\ I_{on2}(V) &= I_{h2}+G_{on2}\cdot(V-V_{h2}). && (19)\end{aligned}$$

Second-snapback-specific parameters — V_{t2}, V_{h2}, I_{h2}, G_{on2}, α_2, β_2 — have essentially the same meanings and extraction procedures as their counterparts for the first snapback. After properly choosing their values and incorporating the second snapback phenomenon, we can fit the measurement data presented in [12] with unprecedented accuracy, as shown in Figure 7.

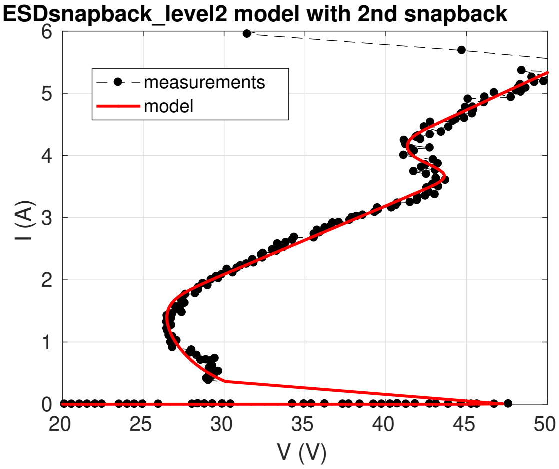

Fig. 7: TLP simulation results of the proposed model with second snapback, overlaid with measurement data.

VI. Incorporating Snapback to Multi-terminal Device Models

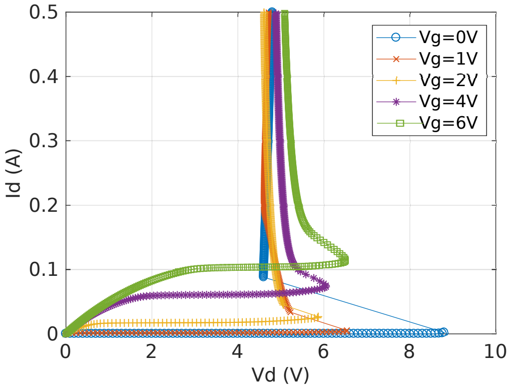

Fig. 8: TLP simulation results of the proposed multi-terminal model.

As a behavioral model for the phenomenon of snapback, the proposed model is not limited to two-terminal devices. As an example, we can incorporate snapback into a four-terminal BSIM3v3 device by tying this model in between the device's drain (d) and source (s) terminals, then making its parameters bias dependent. Specifically, instead of using fixed values for the trigger voltage V_{t1} and the on-state conductance G_{on1}, we make both of them univariate functions of the gate voltage $V(g,b)$:

$$V_{t1} = S_{t1}(V(g,b)), \quad G_{on1} = S_{on1}(V(g,b)), \quad (20)$$

where S_{t1} and S_{on1} are cubic spline interpolation functions, specified by the data points obtained from measurements. Characteristics of such a device are shown in Figure 8, showing realistic snapback behaviors.

Conclusion

In this paper, we discussed the general equation structure to follow when writing robust behavioral models for ESD protection devices. The proposed formalism avoids the use of if-else conditions and instead captures the abrupt turn-on/off behaviors of ESD protection devices using entirely smooth and continuous differential equations. Furthermore, we showed the proper way of modeling such differential equations with internal unknown variables in the Verilog-A language. The presented model based on this formalism is both accurate and robust in circuit simulation. We also discussed the possibilities of extending this generic formalism to incorporate the second snapback phenomenon, as well as applying it to multi-terminal device models.

References

[1] C.H. Diaz and S. Kang. New algorithms for circuit simulation of device breakdown. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 11:1344–1354, 1992.

[2] A. Amerasekera, S. Ramaswamy, M.-C. Chang, C. Duvvury. Modeling MOS snapback and parasitic bipolar action for circuit-level ESD and high current simulations. In *IEEE International Reliability Physics Symposium*, pages 318–326. IEEE, 1996.

[3] E. Li, S.-M. Kang, and others. Circuit-level Simulation of CDM-ESD and EOS in Submicron MOS Devices. In *Electrical Overstress/Electrostatic Discharge Symposium*, pages 316–321. IEEE, 1996.

[4] J. Li, S. Joshi, R. Barnes, E. Rosenbaum. Compact modeling of on-chip ESD protection devices using Verilog-A. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25(6):1047–1063, 2006.

[5] P. A. Juliano, E. Rosenbaum. A novel SCR macromodel for ESD circuit simulation. In *International Electron Devices Meeting*, pages 14–3. IEEE, 2001.

[6] Y. Zhou, D. Connerney, R. Carroll, and T. Luk. Modeling MOS snapback for circuit-level ESD simulation using BSIM3 and VBIC models. In *Sixth international symposium on quality electronic design*, pages 476–481. IEEE, 2005.

[7] Y. Zhou, J. Hajjar, A.W. Righter, and K.P. Lisiak. Modeling Snapback of LVTSCR Devices for ESD Circuit Simulation Using Advanced BJT and MOS Models. In *Electrical Overstress/Electrostatic Discharge Symposium (EOS/ESD)*, volume 29, page 175, 2007.

[8] L. Lou, J. J. Liou. An improved compact model of silicon-controlled rectifier (SCR) for electrostatic discharge (ESD) applications. *IEEE Transactions on Electron Devices*, 55(12):3517–3524, 2008.

[9] R. Mertens, E. Rosenbaum. A physics-based compact model for SCR devices used in ESD protection circuits. In *IEEE International Reliability Physics Symposium (IRPS)*, pages 2B–2. IEEE, 2013.

[10] L. Wei, C.E. Gill, W. Li, R. Wang and M. Zunino. A convergence robust method to model snapback for ESD simulation. In *CAS 2011 Proceedings (2011 International Semiconductor Conference)*, 2011.

[11] N. Monnereau, R. Caignet, N. Nolhier, D. Trémouilles, M. Bafleur. Behavioral-modeling methodology to predict electrostatic-discharge susceptibility failures at system level: An IBIS improvement. In *EMC Europe 2011 York*, pages 457–463. IEEE, 2011.

[12] R. Ida and C.C. McAndrew. A physically-based behavioral snapback model. In *Electrical Overstress/-Electrostatic Discharge Symposium (EOS/ESD)*, pages 1–5. IEEE, 2012.

[13] C. C. McAndrew, G. J. Coram, K. K. Gullapalli, J. R. Jones, L. W. Nagel, A.S. Roy, J. Roychowdhury, A.J. Scholten, Geert D.J. Smit, X. Wang and S. Yoshitomi. Best Practices for Compact Modeling in Verilog-A. *IEEE J. Electron Dev. Soc.*, 3(5):383–396, September 2015.

[14] G. Coram. How to (and how not to) write a compact model in Verilog-A. In *Behavioral Modeling and Simulation, 2004. BMAS 2004. Proceedings of the 2004 International Workshop on*, pages 97–106. IEEE, 2004.

[15] J. Roychowdhury and R. Melville. Delivering Global DC Convergence for Large Mixed-Signal Circuits via Homotopy/Continuation Methods. *IEEE Trans. CAD*, 25:66–78, Jan 2006.

[16] Accellera. Verilog-AMS Language Reference Manual, version 2.4, 2014.

[17] V. Parthasarathy, R. Zhu, V. Khemka, T. Roggenbauer, A. Bose, P. Hui, P. Rodriquez, J. Nivison, D. Collins, Z. Wu, and others. A 0.25μm CMOS based 70V smart power technology with deep trench for high-voltage isolation. In *Proc. IEEE IEDM*, pages 459–462. IEEE, 2002.

[18] Y. Tsividis and C. C. McAndrew. *Operation and Modeling of the MOS Transistor*. Oxford Univ. Press, 2011.

Appendix A
Verilog-A Model Code

```
`include "disciplines.vams"

module ESDsnapback(p, n);
    inout p, n;
    electrical p, n, ns;

    parameter real Vt0 = 45 from (0:inf);    // trigger voltage before the first snapback
    parameter real Vt1 = 50 from (0:inf);    // trigger voltage for the first snapback
    parameter real It1 = 1e-3 from (0:inf);  // trigger current for the first snapback
    parameter real Vh1 = 30 from (0:inf);    // holding voltage for first snapback
    parameter real Ih1 = 1 from (0:inf);     // holding current for first snapback
    parameter real Gon1 = 0.14 from (0:inf); // "on" state conductance after first snapback

    parameter real C = 1e-12 from [0:inf);   // parallel capacitance
    parameter real tau = 1e-8 from (0:inf);  // time constant of impact ionization

    parameter real Is = 1e-12 from (0:inf);  // reverse saturation current of PN junction
    parameter real VT = 0.026 from (0:inf);  // thermal voltage, maybe use $vt
    parameter real VD = 1 from (0:inf);      // "dc leakage early voltage
    parameter real Gmax = 2 from (0:inf);    // maximum conductance under reverse bias

    parameter real K1 = 10 from (1:inf);     // fitting parameter for adjusting the "sharpness" of snapback
    parameter real Alpha1 = 5 from [1:inf);  // fitting parameter for adjusting the shape of snapback
    parameter real Beta1 = 0.1 from [0:inf); // fitting parameter for adjusting the shape of snapback
    parameter real smoothing = 1e-10 from (0:inf); // smoothing factor, normally very small
    parameter real Vsbd = 10 from (0:inf); // tuning parameter for soft breakdown before snapback
    parameter real smoothing_sbd = 1e-3 from (0:inf); // smoothing factor for soft breakdown before snapback

    real s, V_Ion0, Ion, Ki, epsilon, Ioff, maxslope, s_on, correcting_factor, Vstar, sstar;

    analog function real safeexp;
        input x, maxslope;
        real  x, maxslope, breakpoint;
        begin
            breakpoint = log(maxslope);
            safeexp = exp(x*(x <= breakpoint))*(x <= breakpoint) +
                (x>breakpoint)*(maxslope + maxslope*(x-breakpoint));
        end
    endfunction // safeexp

    analog function real smoothclip;
      input x, smoothing;
      real  x, smoothing;
      begin
        smoothclip = 0.5*(sqrt(x*x + smoothing) + x);
      end
    endfunction // smoothclip

    analog function real smoothstep;
        input x, smoothing;
        real  x, smoothing;
        begin
            smoothstep = 0.5*x/sqrt(x*x + smoothing) + 0.5;
        end
    endfunction // smoothstep

    analog begin
        s = V(ns, n);
        V_Ion0 = Vh1 - Ih1/Gon1; // voltage at which Ion is 0
        Ion = smoothclip(Gon1*(V(p, n) - V_Ion0), smoothing) - smoothclip(-Gon1*V_Ion0, smoothing);
        maxslope = VT * Gmax / Is; // maximum slope of safeexp that corresponds to Gmax
        Ki = It1 / (exp(-Vsbd/(Vt1-Vt0))*(Vt1-Vt0));
        epsilon = 0.1; // keep SBD-related smooth functions from operating near 0
        Ioff = Is * (1 - safeexp(-V(p, n)/VT, maxslope)) * sqrt(1 + max(V(p, n), 0)/VD)
              + Ki * smoothstep(V(p, n)-Vt0-epsilon, smoothing_sbd)
              * (smoothclip(V(p, n)-Vt0-epsilon, smoothing_sbd) + epsilon)
              * exp(-Vsbd/(smoothclip(V(p, n)-Vt0-epsilon, smoothing_sbd) + epsilon));
        I(p, n) <+ Ioff + (pow(s, Alpha1)+Beta1*s)/(1+Beta1) * Ion;
        I(p, n) <+ ddt(C * V(p, n));

        s_on = sqrt(1-1/K1);
        correcting_factor = s_on - 1/K1 * atanh(s_on); // ~0.8 for K=10, ~0.9 for K=20
        Vstar = correcting_factor * 2*(V(p, n)-0.5*Vt1-0.5*Vh1)/(Vt1-Vh1);
        sstar = 2*(s-0.5);
        I(ns, n) <+ tanh(K1*(Vstar + sstar)) - sstar;
        I(ns, n) <+ ddt(-tau*s);
    end
endmodule
```

Listing 1: `ESDsnapback.va`