ESD Electronic Design Automation Checks

by ESD Association EDA Working Group

Part II: Implementing ESD EDA Checks in Commercial Tools
by Matthew Hogan

I. Overview

Verification of electrostatic discharge (ESD) design rules has grown in volume and complexity as IC designs have become more complex and added significantly more power domains. With each additional power domain, verification of the signals that cross these domains becomes more difficult (particularly in the identification of inadvertent paths), as well as the check of interactions between circuit blocks that may result in many potential ESD discharge current paths [1]. While not strictly related to ESD, designs that incorporate multiple power domain checks are particularly susceptible to subtle design errors that are difficult to identify in the simulation space or with traditional PV techniques. Often, these subtle reliability errors don't result in immediate part failure, but performance degradation over time. Effects such as Negative Bias Temperature Instability (NBTI) can lead to the threshold voltage of the PMOS transistors increasing over time, resulting in reduced switching speeds for logic gates [2-4], while Hot Carrier Injection (HCI), which alters the threshold voltage of NMOS devices over time, [5] and soft breakdown (SBD) [5] also contribute as time-dependent failure mechanisms, adding to the degradation effects of gate oxide breakdown.

ESD rules for ICs with multiple power domains, IP reuse, and system integration require greater complexity to avoid device damage. Design hierarchy also comes into play where some rules are applied on a top cell and/or top pads, but others are applied between internal blocks that cross multiple power domains. Tracking the rules and the nets to which they apply is by no means a trivial task when performed manually. Automation is necessary to effectively and efficiently cope with these requirements.

As a result, multiple methods have been developed using modeling or simulation to perform chip-level ESD verification [6-8]. However, while simulation-based ESD verification methods, to verify compliance to human body model (HBM) and charged device model (CDM) requirements, are effective, they do not necessarily check all elements in the design for ESD violations. In particular, internal interfaces between different supply domains are not explicitly checked. Additionally, getting device models for simulation at these extreme conditions is often problematic.

Part I of this series, "Part I: Outlining the Essential requirements of the ESD Verification Flow" provided an overview of the essential requirements of an effective ESD EDA verification flow [14]. This article (Part II) discusses a well-established topological methodology for checking ESD design rules. The ESDA technical report 18, "ESD Electronic Design Automation Checks" (TR18) [13], provides an overview of recommended ESD checks that should be performed to validate appropriate ESD protection structures within a design. We will focus our effort on TR18 rule 5.1.3, which applies to internal interfaces between power or ground domains, a requirement that has been recently highlighted [9-11]. Rather than modeling or simulating, the methodology uses the device netlist topology to check all domain crossing interfaces and associated ESD devices in the entire design, and is realized using the Calibre® PERC™ tool from Mentor Graphics. Although internal interfaces may span many levels in the design hierarchy,

checking is done hierarchically, utilizing a novel technique for topology-aware verification. In addition to only performing topology checking, at times there is the need to include both topology and physical information to create a more comprehensive checking environment. Such an environment is required to perform ESD layout verification checks [12].

The following sections cover the targeted ESD rules (section II), the new hierarchical algorithm (section III), ESD rule variations (section IV) and verification results (section V).

II. The ESD Rule

Transistors' gates can be exposed to direct ESD events. This is particularly common in input receivers, although many other topologies can expose a gate oxide to an ESD discharge path. Since gate oxides (by virtue of their small capacitance) cannot shunt any significant amount of current, they have to be considered voltage pulse driven, as far as their failure mechanism is concerned. It is irrelevant whether the gate oxide is connected to signal, ground, or supply. The failure criteria will depend on the actual combination exercised, and whether a soft vs. hard oxide breakdown sets the failure limit (application-dependent) [13].

ESDA TR18, check 5.1.3 [13] is intended to verify presence of protections on signals that cross a power domain boundary. As shown in Figure 1, when the pad VDD1 is struck with respect to VSS2, a high voltage could be developed across the gate-source oxide of the NMOS in the VDD2 power domain.

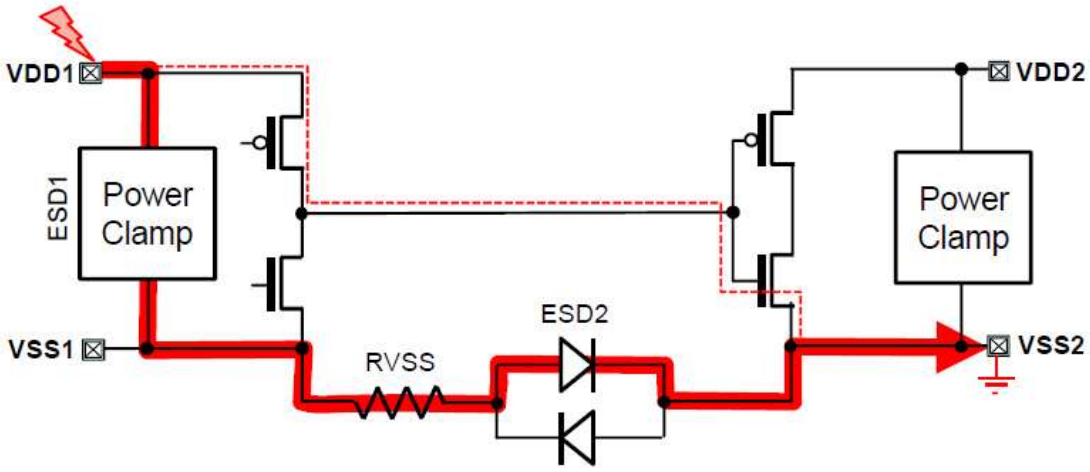

Figure 1. Typical Signal Cross-Domain ESD Issue *(source: EDA Tool Working Group (2011). ESD Electronic Design Automation Checks (ESD TR18.0-01-11)[13]*

To define our rule, we begin by identifying the ESD protection strategy: to protect this component we need to ensure that the voltage across it does not exceed the set failure level. A simplified overview of the check that needs to be performed to ensure the gate oxide is adequately protected is as follows:

```
For each net in design,
  IF net connects driver and receiver THEN
     check power domains of driver and receiver
      IF different power domains THEN
```

```
check for anti-parallel diodes
IF anti-parallel diodes do not exist THEN
   ESD error
```

Drivers and receivers are determined by net connectivity, as are the different power domains. Because this is an interface, the pieces of the circuit that must be checked are usually distributed between different levels of the design hierarchy, so it is not obvious how to check the rule independently on a cell-by-cell basis. However, using a flat approach does not provide sufficient capacity to run larger chips. For scalability reasons, it becomes necessary to develop a hierarchical topological approach to efficiently solve this issue. In the next section, we present such a method that performs hierarchical verification.

III. Hierarchical Verification

Overview

The first requirement is a SPICE netlist, which can be either a schematic netlist or a netlist extracted from the layout. In the latter case, the LVS-like runset used for extraction must ensure that all ESD protection devices are extracted (Note: parasitics are not extracted, just intentional devices). While the netlist must contain the proper text names for device pins (so that power and ground domains can be established), in general, texting in the netlist is not used extensively for verification (see Figure 2).

The second input is an ESD rule deck. It specifies the ESD design rules to be checked, and the list of power and ground domain names. Power and ground names are not generated automatically; they must be specified in the rule deck per the design specification. This rule deck is essential for making the verification method generic. For ease of discussion, however, we will describe the method in the context of the ESD design rule formulated in Section II.

Conceptually, the hierarchical algorithm runs in two steps: 1) initialization, and 2) rule checking. In the initialization step, the algorithm gathers ESD-related topology information from each cell, and propagates it throughout the design. In the second step, the algorithm checks ESD design rules independently, cell by cell, as each cell now has access to the entire ESD protection scheme propagated from all other cells.

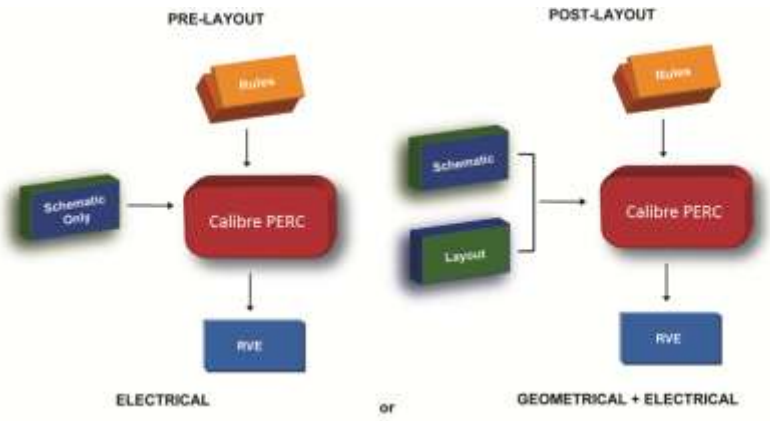

Figure 2. Hierarchical verification flow

ESD Rule Checking

Once net connectivity is defined, we can check the ESD design rule cell by cell. Since a net's path through devices is, in general, instance-dependent, we cannot just check each cell once. Instead, we find a list of representative instances with unique net connectivity for each cell. Depending on the amount of regularity in the design, the list of instance representatives can be orders of magnitude smaller than the list of all instances for a cell. This greatly improves the speed of the tool compared to checking a flat netlist, and is done while preserving any instance specific configurations.

Rule Deck Coding Considerations

Given the diversity in ESD rules, it is important to develop a robust rule deck that will not miss real violations. Within the framework of our method, there are two basic approaches: one is to code a new rule for each variation, and the other is to code a single general purpose rule that covers all variations. The tradeoff is speed vs. rule complexity. The first approach is simpler, but slower, as each net will be checked multiple times (once for each rule). The second approach is faster, but obviously more complex.

The rules should include checking of properties of the ESD protection devices, such as such as ESD components widths. Also, the rules should handle different protection types. For example, the ESD protection circuit in Figure 1 could be a dynamic or static clamp or diodes.

Similarly, the drivers and receivers in real circuits are not necessarily simple inverters. They can be NANDs, NORs, etc. However, this does not need special attention from the rule writing point of view. The tool automatically handles different types of logic gates.

Moreover, the tool can recognize multiple drivers/receivers on an interface net—for instance, a driver with a fan-out to three inverters (in the same domain or in different domains). The rules should take advantage of this ability, and report all drivers/receivers associated with a violation.

At the global level, a robust rule deck should also include other ESD checks. For example, the parameters used in the domain crossing interface check can be dependent on properties of the supply protections. As an example, in the case shown in Figure 1, where the driver and receiver have separate VDDs and VSSs, we are able to make a determination of the checks to be performed and determine the need for the specific protection circuit specified (in this case, anti-parallel diodes).

IV. Results

ESD rule decks have been written using this technique and have been verified in production design flows for both large blocks and complete chips. We'll review the results in terms of functionality (how well did it identify real problems?) and reporting (how easy is it for users to manage and correct errors?).

Functionality

In practice, designs with multiple power and ground domains often involve hundreds or thousands of crossings that need to be verified. In addition to determining what signals require ESD clamps for protection, the crossing audit is also needed to determine which ground domains need interface protections.

In one example, for noise isolation purposes, a PLL was designed with separate ground domains for the core and 1.8V circuits. Traditionally, crossings between domains were checked manually to see if ESD clamps were present. However, crossings can be very difficult to find, since the connections may need to be traced through multiple schematics, and there can be hundreds, if not thousands, of crossings. Using the PLL example, the hierarchical ESD audit identified all 133 crossings in just a few seconds. The crossing audit also successfully caught missed instances of clamps in the preliminary design.

Reporting

The output from the rule deck lists all the crossing nets and is organized by hierarchy (Figure 3). For each net, the MOSFETs on both sides of the interface, together with the associated grounds, are shown. This output can be customized as desired, and Calibre PERC provides a results viewing environment (Calibre RVE) to highlight devices in the schematic and/or layout when they are selected in the report. All 133 results are displayed in the graphical tree view shown in Figure 3. Analysis of these results will identify the specific details for each failure.

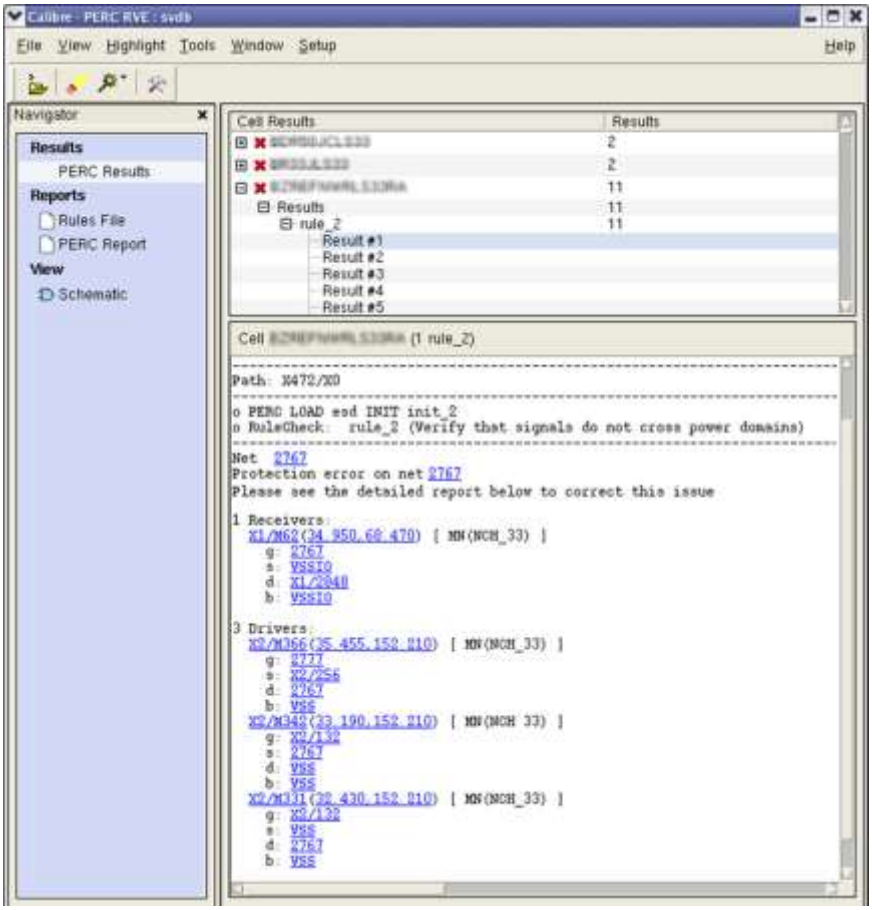

Figure 3. Results for entire design, showing an ESD protection error on net 2767, involving one receiver and three drivers.

The schematic view in the results viewer can provide a different view of this error (Figure 4). This often provides a holistic view of the connectivity enabling much easier debugging than the original schematic. Of course, as these results are displayed in Calibre RVE, highlighting back to the original schematic is also supported.

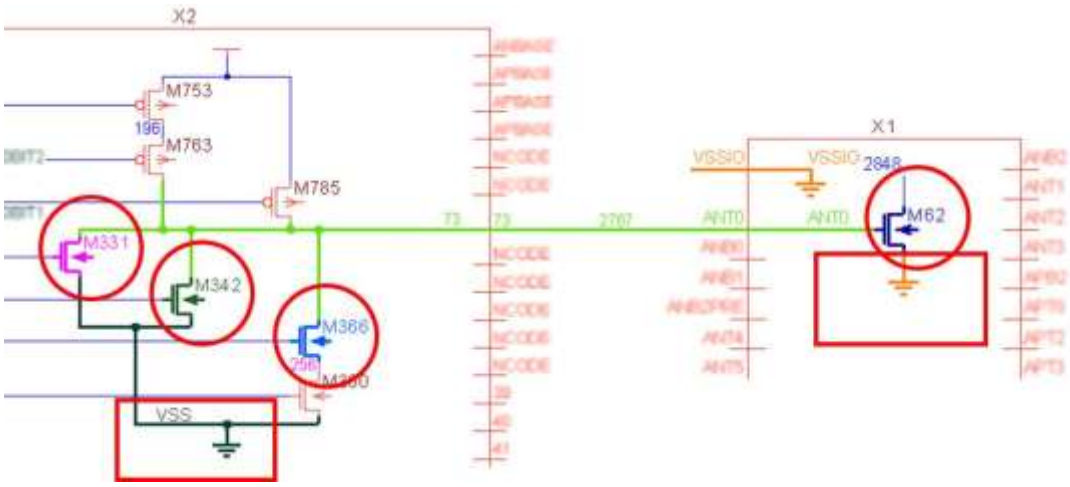

Figure 4. Schematic view from results viewing, identifying all the circuitry elements affected by the error: Net 2767, Receiver: X1/M62, Drivers: X2/M331, X2/M341, X2/M366, Ground nets VSS and VSSIO.

Because you can specify nets, devices, pins, etc., and create “groupings” for testing conditions, the tool can use these conditions to determine how to evaluate the design.

V. Conclusion

In this paper, we presented a well-established topologically driven hierarchical verification methodology that has been developed to automate ESD rule checking. It can handle large ICs, and check ESD protection rules on the original design without netlist reduction. The hierarchical algorithm uses a novel topology-aware concept, allowing for verification of chip-level ESD design rules. The presented method has been extensively verified, and is being used in production to significantly improve ESD quality.

Until now, there has been a clear gap in EDA solutions to address the demands of circuit and electrical verification. The ability to use both netlist and layout (GDS) information simultaneously to perform electrical checks enables designers to address both reliability concerns arising from crossing multiple power domains and catastrophic failures from ESD that can have large effects on yield and reliability. In addition, this method can employ topological constraints to verify that the correct structures are in place wherever circuit design rules require them. An automated solution that verifies circuits at both the schematic and layout phase can reduce cost and time to market, while improving yield and device reliability.

At the time of writing the ESDA EDA Working Group consisted of the following members: Michael Khazhinsky (Silicon Labs), Fabrice Blanc (ARM), Gianluca Boselli (Texas Instruments), Shuqing (Victor) Cao (Global Foundries), Norman Chang (Ansys), Dan Clement (On Semiconductor), Rosario Consiglio (Impulse Semiconductor), Maxim Ershov (Silicon Frontline), Melanie Etherton (Freescale Semiconductor), Eleonora Gevinti (ST), Harald Gossner (Intel), Matthew Hogan (Mentor Graphics), Larry Horwitz (Synopsys), Kelvin Hsueh (ESD Consultant), Mujahid Muhammad (IBM), Louis Thiam (Cadence), Nitesh Trivedi (Infineon), Vesselin Vassilev (Novorell).

References

[1] M. Ker, C. Wu, H. Chang, and T. Wu. "Whole Chip ESD Protection Scheme for CMOS Mixed- Mode IC's in Deep-Submicron CMOS Technology". In Proceedings of the IEEE Custom Integrated Circuits Conference, 1997, pp. 31-34.

[2] Hamed Abrishami, et al., "NBTI-Aware Flip-Flop Characterization and Design", GLSVLSI'08, May 4–6, 2008

[3] B.C. Paul, K. Kang, H. Kuflouglu, M. A. Alam and K. Roy, "Impact of NBTI on the temporal performance degradation of digital circuits," Electron Device Letter, vol. 26, no. 8, pp. 560-562, Aug. 2005.

[4] Hong Luo, et al., "Modeling of PMOS NBTI Effect Considering Temperature Variation", 8th International Symposium on Quality Electronic Design (ISQED'07)

[5] Jin Qin, et al., "SRAM Stability Analysis Considering Gate Oxide SBD, NBTI and HCI", 2007 IIRW FINAL REPORT

[6] J. Lee, K. Kim, and S. Kang. "VeriCDF: A New Verification Methodology for Charged Device Failures." In Proceedings of the 39th Design Automation Conference, 2002, pp. 874-879.

[7] H. Qian, J.N. Kozhaya, S.R. Nassif, and S.S. Sapatnekar. "A Chip-level Electrostatic Discharge Simulation Strategy." In Proceedings of the IEEE/ACM International Conference Computer Aided Design, 2004, pp. 315-318.

[8] M. Baird and R. Ida. "VerifyESD: A Tool for Efficient Circuit Level ESD Simulations of Mixed-Signal ICs." In Proceedings of the ESO/ESD Symposium, 2000, pp. 465-469.

[9] Willemen, Joost and Cerati, Lorenzo. May, 2012. "ESD design in high voltage technologies". Private Communication, May 14, 2012.

[10] Shih-Hung Chen, Ming-Dou Ker, Fellow, IEEE, and Hsiang-Pin Hung, "Active ESD Protection Design for Interface Circuits Between Separated Power Domains Against Cross-Power-Domain ESD Stresses", IEEE Transactions On Device And Materials Reliability, Vol. 8, No. 3, pp 549-560. September 2008.

[11] J. Lee and E. Rosenbaum, "Voltage clamping requirements for ESD protection of input in 90nm CMOS technology," Proc. of 30th EOS/ESD Symposium, pp. 50-58, 2008.

[12] T. Smedes, N. Trivedi, J. Fleurimont, A.J. Huitsing, P.C. de Jong, W. Scheucher, and J. van Zwol. "A DRC-based check tool for ESD layout verification." In Proceedings of the ESO/ESD Symposium, 2009.

[13] EDA Tool Working Group (2011). ESD Electronic Design Automation Checks (ESD TR18.0-01-11). New York: Electrostatic Discharge Association. http://www.esda.org/Documents.html

[14] Michael Khazhinsky, "ESD Electronic Design Automation Checks." InCompliance Magazine, August 2012