A Chip-level Electrostatic Discharge Simulation Strategy

Haifeng Qian
Univ. of Minnesota
Minneapolis, MN
qianhf@ece.umn.edu

Joseph N. Kozhaya
IBM Corp.
Essex Junction, VT
kozhaya@us.ibm.com

Sani R. Nassif
IBM Corp.
Austin, TX
nassif@us.ibm.com

Sachin S. Sapatnekar
Univ. of Minnesota
Minneapolis, MN
sachin@ece.umn.edu

Abstract

This paper presents a chip-level charged device model (CDM) electrostatic discharge (ESD) simulation method. The chip-level simulation is formulated as a DC analysis problem. A network reduction algorithm based on random walks is proposed for rapid analysis, and to support incremental design. A benchmark with a 2.3M-node V_{DD} net and 1000 I/O pads is checked in 13 minutes, and 10 re-simulations for incremental changes take a total of 9 minutes.

1. INTRODUCTION

Electrostatic discharge (ESD) is an important issue during the manufacture of a chip: when the chip contacts with an assembly-line probe or a human body, a surge of discharge current could cause permanent damage. The product losses due to ESD were reported to be 16-22% in electronic component manufacturing as early as 1990 [3]. As feature sizes reduce, thinner gate oxides come into use, and design complexity grows, circuits are increasingly vulnerable to ESD damage. To protect against this problem, a modern design usually employs a full-chip ESD protection strategy. Figure 1 illustrates a schematic that contains internal functionality circuitry and ESD protection devices [5][11][14]: *I/O protection* circuitry diverts discharge current from an I/O pad into the V_{DD} net and/or the ground net; *ESD voltage clamps* provide paths for the charge to be drained from the V_{DD} net into the ground net; *Diode strings* (not explicitly shown in Figure 1) are placed between the different power nets. The ESD discharge path may correspond to either the primary power supply net, or the secondary supplies for I/O circuitry. We generically refer to each of these two as the V_{DD} net.

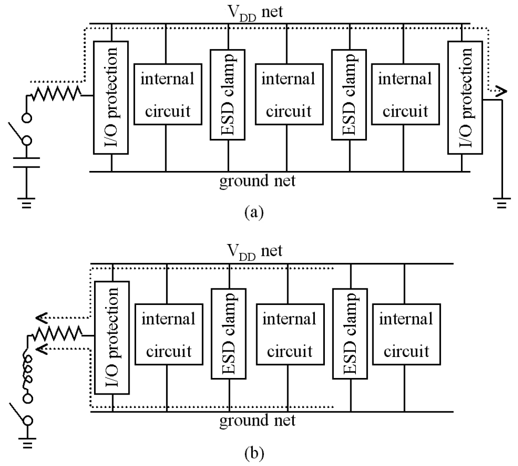

Figure 1. ESD simulations: (a) an HBM event (b) a CDM event. Dotted lines represent desirable discharge paths.

Before manufacturing, simulations must be carried out to determine whether these protective devices are adequate to withstand a specified level of ESD stress, by imitating physical tests regulated by industrial standards, e.g., [4]. Two widely used models are:

- Human body model (HBM): an external capacitor is discharged through a pair of pads, as shown in Figure 1(a).
- Charged device model (CDM): the charge accumulated on the chip itself goes through a single grounded pad, as shown in Figure 1(b). It was reported that CDM accounts for a majority of ESD damage during chip manufacturing [7].

SPICE-like models and techniques have been proposed for circuit-level ESD simulation [1][9][10], simulating one I/O protection at a time. As the design complexity grows, especially for chips with multiple power domains, unexpected discharge paths often cause failures that are not visible in circuit-level simulations, and consequently methodologies have been proposed to address chip-level ESD simulation [7][8][15]. The large problem size prohibits a full SPICE-like simulation, and these efforts all apply techniques to reduce computation. For example, [7], targeted at CDM simulation, builds a macromodel for each power domain, where the charge source is represented by a pair of lumped capacitors, and performs detailed transient analysis for the reduced full-chip model.

This paper presents a different approach for simulating CDM events. The chip-level simulation is formulated as a DC analysis problem of finding the voltage at a stressed I/O node. This voltage is used as an indicator of potential ESD failure. To facilitate the DC analysis that must be carried out for each I/O pad, a network reduction method based on random walks is proposed, with the desirable property that when a change is made in the design to fix an ESD violation, re-simulation can be performed by local computations without resolving the whole circuit.

2. PROBLEM FORMULATION

The purpose of chip-level CDM simulation is to compute the voltage drop along the discharge paths. A strong correlation has been established between hardware ESD failures and the wire resistance of the discharge paths [2][15], and may be explained as follows. ESD voltage clamps, which constrain the voltage difference between the V_{DD} net and the ground net, are only fully effective at the points where they are connected. For I/O pads placed far away from ESD clamps, in a CDM event, the discharge current flows through the V_{DD} grid, and causes a voltage difference that is proportional to the wire resistance of the discharge path. When the voltage on the V_{DD} net exceeds a certain threshold, an ESD failure may be induced.

A DC formulation that captures the required chip-level CDM simulation is illustrated in Figure 2. The circuit is modeled as follows:

- **The V_{DD} net** is extracted from the layout and modeled as a resistive network. The resistance of ground net is ignored in this model. The reason is that the V_{DD} net corresponds to the secondary power net for I/O or the power net of a single power domain, and this typically has higher resistance than the global ground net that serves all power domains, and is the lowest impedance net on chip. If the ground net has a significant resistance, it can be modeled in a similar manner.
- **An ESD event at an I/O pad** is modeled as a current source placed at the location of that I/O. The value of this

current source is defined to be the peak CDM surge current specified in the JEDEC standard [4].
- **An ESD voltage clamp** is modeled as a voltage source in series with a resistor, placed at its physical location. The values of the voltage source and the resistor are obtained from simulated I-V curves of clamps under a stressed situation.

In the above model, the I/O pads are simulated one at a time, under the assumption that discharge occurs at only one I/O pad. Thus, the number of simulations equals the number of pads. For the j^{th} simulation, a current value is assigned for the current source modeling the j^{th} I/O pad, zero current for all the other current sources. The computed voltage at the j^{th} I/O power node is checked against the allowable threshold V_{limit} to determine whether the ESD specification is met. The simulation is repeated for all I/O pads. Note that in every simulation, the voltage source-resistor elements (ESD clamp models) are all active, since all clamps help in providing a discharge path to the ground network. The problem size is much larger than what is shown in Figure 2: the V_{DD} net can have up to millions of nodes; the number of I/O pads may be up to a few thousands; there are typically 30 to 40 ESD clamps, depending on the physical and electrical constraints of the chip.

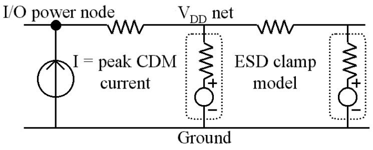

Figure 2. A DC model for chip-level CDM simulation of the circuit shown in Figure 1(b).

During the simulation, if the threshold V_{limit} is exceeded at an I/O, this I/O is considered a potential ESD failure, and one of the following methods may be used to fix this violation:

- Reduce the effective resistance of the discharge paths by using wider wires to connect this I/O to power grid, or move the I/O circuitry closer to a power bus.
- Add an ESD clamp at a nearby location.

Such design changes require incremental re-simulations, and it is important for the analysis to be able to do so rapidly.

Although an ESD event is fundamentally a transient phenomenon, the DC formulation is justified by being conservative, using the peak of the CDM current waveform as the input excitation, while brings the benefit of a much faster simulation, as compared to a complete transient analysis. The computed voltage at the V_{DD} node of the stressed I/O represents the worst-case voltage drop along the discharge paths, and has been shown to be a good indicator of potential ESD failure: in a 90nm ASIC, I/O failures start to occur when the path voltage drop exceeds 13V [2]. Different I/O pads may be checked against different thresholds depending on their designs.

3. NETWORK REDUCTION

The computational complexity of the DC formulation can be still high, with the V_{DD} net containing up to several millions of nodes, and the number of simulations being equal to the number of I/O pads, up to a few thousands. To further speed up, we note that the number of current and voltage sources is limited. Therefore, it is desirable to perform a network reduction and build an equivalent circuit that only contains the nodes with a current or voltage source. Then DC analysis can be carried out for this reduced circuit only, with one current source being on during each simulation.

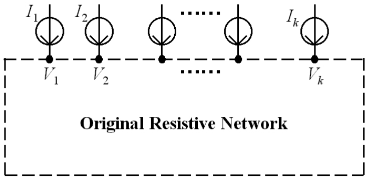

Figure 3. The original resistive network with external connections replaced by current sources.

Figure 3 illustrates the resistive network to be reduced. It is composed of resistors, and a number, k, of its nodes have external connections, each of which, in the ESD context, corresponds to an I/O protective device or ESD clamp model. In general, there may be constant current sources between internal nodes and ground. By the terminology of [16], we refer to the k nodes that have external connections as ports, with port voltages $\mathbf{V_{ports}} = [V_1, V_2, \cdots, V_k]^T$, and we denote the port currents injected into the network by $\mathbf{I_{ports}} = [I_1, I_2, \cdots, I_k]^T$. In Figure 3, the external connections are replaced by symbolic current sources. We can do this because $\mathbf{V_{ports}}$ and $\mathbf{I_{ports}}$ are treated as algebraic symbols throughout this section, and the equations apply to all possible values of them. The goal of network reduction is to find a square matrix A and a constant vector $\mathbf{S}$ such that the following equation holds for all possible $\mathbf{V_{ports}}$ and $\mathbf{I_{ports}}$.

$$\mathbf{I_{ports}} = A\mathbf{V_{ports}} + \mathbf{S} \tag{1}$$

For a connected network, the exact A is a full matrix, i.e., the exact reduced circuit is a clique, which will not result in any runtime advantage. Therefore, an algorithm is needed to produce an estimated A matrix with reasonable sparsity, without excessive loss of accuracy.

The proposed method is a modified version of the virtual-layer algorithm from [13], to achieve a much lower reduction rate than the 10% rate recommended by [13] to ensure connectivity. [12][13] construct a random walk "game" to model a resistive network, and estimate a node voltage by performing a number of walks from that node and computing the average "gain" in those experiments. The terms "motel," "home" and "award" are as defined in [12], and the formal definitions are omitted here. In the proposed algorithm, a symbolic estimation of port voltage V_i, is obtained by setting up the game as follows:

- A set of M walks are run from a port i, $i \in \{1, 2, \cdots, k\}$, inside the network shown in Figure 3.
- The motel price at port i is $\frac{-I_i}{\sum_{t=1}^{\mathrm{d}(i)} g_t}$, where I_i is the port current, $\mathrm{d}(i)$ is the number of resistors connected to port i inside the network, and g_t's are the conductances of these resistors. Note that I_i is symbolic, and any computation regarding this motel price is carried out symbolically.
- All of the ports are home nodes where random walks end, except for port i itself. We refer to these ports as *absorbing* nodes, while port i and non-port nodes are *non-absorbing* nodes. In other words, a random walk cannot end at port i, and has to reach a port other than i to stop. The award for reaching a port is its port voltage. We do not know these values, and computation is carried out symbolically.
- Port currents other than I_i are at absorbing nodes, and do not generate motels; each constant current source inside the network becomes a motel with the price $\frac{I_{\mathrm{source}}}{\sum g}$, where I_{source} is the current flowing from an internal node to ground, $\sum g$

is the sum of conductances connected to that node.

For each individual walk in the above game, the money earned at the end of the walk is composed of an award, which is a port voltage, minus a sequence of motel expenses, in the following form:

$$W_q = V_{\mathrm{end}\,q} + r_q \frac{I_i}{\sum_{t=1}^{\mathrm{d}(i)} g_t} - c_q \qquad (2)$$

where $q \in \{1, 2, \cdots, M\}$ is the index of the walk, $V_{\mathrm{end}\,q}$ is the port voltage at the end of the walk, r_q is the number of times that the walk passes port i, c_q is the total expense paid at motels corresponding to internal current sources. Note that c_q is a constant number, i.e., it is independent of $\mathbf{I_{port}}$ and $\mathbf{V_{port}}$. Taking the average of the M results, an estimated V_i is obtained as follows.

$$V_i = \frac{\sum_{q=1}^{M} W_q}{M} = \sum_{\substack{j\in\{1,\cdots,k\}\\ j\neq i}} \frac{N_j}{M} V_j + \frac{R}{M}\frac{I_i}{\sum_{t=1}^{\mathrm{d}(i)} g_t} - C(3)$$

$$\text{where} \qquad R = \sum_{q=1}^{M} r_q \qquad C = \frac{\sum_{q=1}^{M} c_q}{M}$$

and N_j is the number of walks that end at port j. R is the total number of times that walks pass port i. C is a constant number independent of $\mathbf{I_{port}}$ and $\mathbf{V_{port}}$. Because every random walk stops at a port that is not port i, the N_j's must satisfy:

$$\sum_{j\in\{1,\cdots,k\},\, j\neq i} N_j = M \qquad (4)$$

By algebraic transformations, equation (3) can be converted into

$$I_i = \frac{M}{R}\sum_{t=1}^{\mathrm{d}(i)} g_t V_i - \sum_{\substack{j\in\{1,\cdots,k\}\\ j\neq i}} \frac{N_j}{R}\sum_{t=1}^{\mathrm{d}(i)} g_t V_j + \frac{MC}{R}\sum_{t=1}^{\mathrm{d}(i)} g_t \quad (5)$$

Comparing equation (5) and equation (1), it can be seen that (5) estimates the i^{th} row in matrix A and the i^{th} entry in vector $\mathbf{S}$ as:

$$A_{i,i} = \frac{M}{R}\sum_{t=1}^{\mathrm{d}(i)} g_t,\ A_{i,j} = -\frac{N_j}{R}\sum_{t=1}^{\mathrm{d}(i)} g_t,\ S_i = \frac{MC}{R}\sum_{t=1}^{\mathrm{d}(i)} g_t \quad (6)$$

So far we have estimated the entries in A and $\mathbf{S}$ that correspond to a specific port i. For each port, we repeat the procedure and construct the matrix A row by row, and the vector $\mathbf{S}$ entry by entry. Equation (6) shows that the estimated matrix A has positive diagonal entries, non-positive off-diagonal entries, and using equation (4), the following equation can be easily proven.

$$\sum_{j=1}^{k} A_{i,j} = 0 \qquad (7)$$

The reduced resistive network becomes easier to visualize when we rewrite equation (5) in the following form:

$$I_i = S_i + \sum_{j\in\{1,\cdots,k\},\, j\neq i} (-A_{i,j})(V_i - V_j) \qquad (8)$$

Following the terminology from [13], equation (8) is an "imaginary circuit" in which $(-A_{i,j})$ conductance connects port i to port j, and an independent current source S_i flows out of node i.

The sparsity-accuracy tradeoff of the reduced network is controlled by M, the number of random walks used. In equation (6), $A_{i,i}$ can be viewed as the total conductance from port i to other ports, and this amount is distributed among $(-A_{i,j})$'s such that each of them gets a portion N_j/M. Hence, M can be considered as the resolution of our estimation. When M increases, the matrix A becomes denser and closer to the exact matrix, and is more expensive to compute.

Finally, for CDM simulation where there are no current sources inside the network, c_q in (2), C in (3), and vector $\mathbf{S}$ are all zero.

4. SIMULATION AND RE-SIMULATION

With a reduced network from the previous section, the I/O and clamp models can now be added as external connections, one current source being on at a time. Any linear solver can perform the DC analysis; we continue to use the framework of [12], because it can estimate one single node voltage without solving the whole circuit. To further reduce runtime, we note that in the ESD simulation, only nodes with high voltages are of interest. Hence the proposed algorithm employs adaptive error margins. The error margin Δ is a parameter in [12] to control the number of random walks, defined as $P[-\Delta < \text{error} < \Delta] > 99\%$. In our implementation, we use three error margins, $\Delta_1 < \Delta_2 < \Delta_3$, and define two thresholds $V_{T1} < V_{T2}$. When estimating an I/O node voltage, the computation starts with Δ_3. After this accuracy level is achieved, if the estimated voltage is below V_{T1}, the computation stops; otherwise, the error margin is changed to Δ_2, and the computation continues. After the new accuracy level is achieved, if the estimated voltage is below V_{T2}, the computation stops; otherwise, the error margin is changed to Δ_1, and the computation continues. Using this adaptive strategy, more runtime is spent on high-voltage nodes, to get more accurate voltage values, while safe I/O nodes only get coarse estimation.

As discussed in the Section 2, if the threshold V_{limit} is exceeded at an I/O, this I/O is considered a potential ESD failure, and the design is modified to fix it. Then a re-simulation is needed to ensure that the I/O node voltage is reduced to a satisfactory level. This can be performed by rerunning the entire process of network reduction and voltage computation, which is expensive. Instead, we propose to locally update the reduced circuit, as follows:

- If wider, or shorter (by placing the I/O circuitry closer to a power bus), wires are used to connect the target I/O to the power grid, we run M walks from this target I/O, and re-compute its connections in the reduced circuit. Then the ports adjacent to this I/O are updated as well, each with M additional walks.
- If an ESD clamp is added at a location nearby the target I/O, this clamp becomes a new port in the original network, and a new node in the reduced circuit. We first run M walks from this new port, and compute its connections in the reduced circuit. Next, the ports that are found to be adjacent to this new port are updated as well.

In the above local updates, we save computation by ignoring possible effect on ports that are not connected to the improved I/O node or to the new ESD clamp. In other words, if random walks from port i never visit port j, we assume that walks from port j also never visit port i. Because of the fact that the connections between two nodes in the reduced circuit are most likely mutual and with similar conductances, the error induced by the local updating is minimal.

Finally, the algorithm in sections 3 and 4 is not limited to CDM simulation, and is applicable to any problem where only a small portion of the nodes in a resistive network are of interest. For example, it can also speed up HBM simulation by reducing every involved resistive network to only the nodes that tie to ESD protective devices.

5. RESULTS

Three benchmarks, described in Table 1, are used for evaluation. The first two are created by randomly assigning I/O pads and ESD clamps on two industrial power grid models. The third benchmark is generated based on the structure of the first benchmark, and with randomly assigned I/O's and clamps. Computations are carried out on a Linux workstation with a 2.8GHz Pentium4 CPU.

The results reported in Table 1 and Table 2 are from simulations with the following parameters:$M = 1000$, $\Delta_1 = 0.1\text{V}$, $\Delta_2 = 0.5\text{V}$, $\Delta_3 = 1\text{V}$, $V_{\text{limit}} = 13\text{V}$, $V_{T1} = 12\text{V}$, $V_{T2} = 12.5\text{V}$. Results for the first two benchmarks are compared against the SPARSE linear solver [6], while the third benchmark is too large for SPARSE.

Table 1. Benchmarks and runtimes. N1 is the number of nodes, N2 is the number of I/O pads, N3 is the number of ESD clamps, T1 is the runtime of an initial complete simulation, and T2 is the runtime of 10 re-simulations afterwards. RW denotes the proposed algorithm.

Ckt	N1	N2	N3	T1		T2	
				RW	SPARSE	RW	SPARSE
#1	36K	500	20	23.28sec	7.6min	49.69sec	61min
#2	101K	700	40	89.30sec	55min	61.65sec	8.5hour
#3	2.3M	1000	40	13min	–	9min	–

Table 2. Accuracy of the initial complete simulation.

Ckt	#1	#2
Voltage range(V)	10.35–14.40	5.61–45.13
Average error(V)	0.19	0.29
Max error(V)	0.80	1.38
Fraction of failures covered	9/9	2/2
Number of false alarms	5	0

Table 3. Runtime-accuracy tradeoff. E1 is average error, E2 is max error. T1 and T2 are as defined in Table 1.

Ckt	#1			#2		
M	1000	3000	10000	1000	3000	10000
E1(V)	0.19	0.10	0.05	0.29	0.14	0.08
E2(V)	0.80	0.38	0.24	1.38	0.57	0.47
T1	23.28sec	69.37sec	3.9min	89.30sec	5.5min	26.7min
T2	49.69sec	3.2min	14min	61.65sec	3.8min	22min

Table 4. Accuracy of re-simulations for the first two circuits: voltage changes at I/O pads that are improved.

Ckt #1					Ckt #2				
	Initial(V)		Final(V)			Initial(V)		Final(V)	
	Exact	Est.	Exact	Est.		Exact	Est.	Exact	Est.
1	13.76	13.60	10.31	10.23	1	45.13	45.30	8.46	8.23
2	14.40	14.46	10.95	10.88	2	13.27	13.23	11.04	10.88
3	12.73	12.84	9.28	9.26	3	12.60	12.50	11.69	11.54
4	13.25	13.27	9.80	9.79	4	12.56	12.47	11.54	11.61
5	13.16	13.26	11.96	12.24	5	11.49	11.48	10.48	10.40
6	13.37	13.18	9.91	9.85	6	11.11	11.16	10.78	10.66
7	12.99	13.11	9.61	9.51	7	11.21	11.10	10.43	10.47
8	12.93	13.05	9.50	9.46	8	11.35	11.20	10.63	10.63
9	13.42	13.56	9.99	10.01	9	11.13	11.10	10.77	10.70
10	12.99	13.00	9.60	9.62	10	11.11	11.01	10.45	10.34

In viewing Table 2, the reader is reminded that due to the adaptive error margins, most high errors occur at the low-voltage safe nodes. Even with these errors, for the voltage range of ESD simulation, the accuracy is sufficient for detecting potential failures. Being conservative, the algorithm reports a failure when an estimated I/O voltage exceeds 12.9V. As shown in the last two rows of Table 2, all real failures reported by SPARSE are covered by our algorithm; as a cost of being conservative, five false alarms are given for the first benchmark, at nodes with voltages below but close to V_{limit}. When desired, higher accuracy can be achieved by increasing M, the number of walks used in network reduction. Table 3 shows the accuracy-runtime tradeoff when M is increased to 3000 and 10000. The error margins are shrunk accordingly in generating the results. Table 4 shows re-simulation results using the local update method from Section 4. (Runtimes are in Table 3.) 10 nodes are chosen arbitrarily in each circuit, and are not limited to those violating the 13V threshold. A design change is made to improve each node sequentially, and a re-simulation is preformed after each change. The voltages before and after the design change at the target I/O node are listed in each row, using both SPARSE and the proposed algorithm, at a high-accuracy setting with $M = 10000$. The most dramatic change is the first node of the second circuit, due to an I/O being assigned at a node with poor connection to major power bus, which emulates the scenario of a poorly designed I/O protection. It is fixed by adding a large via from this I/O to a major power wire, and our local update method captures the corresponding voltage change.

REFERENCES

[1] S. G. Beebe. Simulation of complete CMOS I/O circuit response to CDM stress. *Proc. EOS/ESD*, pp. 259-270, 1998.

[2] C. Brennan, J. Kozhaya, R. Proctor, J. Sloan, S. Chang, J. Sundquist and T. Lowe. ESD design automation for a 90nm ASIC design system. To appear in *Proc. EOS/ESD*, 2004.

[3] *Fundamentals of ESD.* ESD Association, Rome, NY, 2001.

[4] *JEDEC Standard No. 22-C101-A: Field-Induced Charged-Device Model Test Method for Electrostatic-Discharge-Withstand Thresholds of Microelectronic Components.* JEDEC Solid State Technology Association, Arlington, VA, 2000.

[5] M.-D. Ker and H. Chang. Whole-chip ESD protection strategy for CMOS IC's with multiple mixed-voltage power pins. *Proc. Int'l Symposium on VLSI Technology, Systems and Applications*, pp. 298-301, 1999.

[6] K. S. Kundert and A. Sangiovanni-Vincentelli. SPARSE 1.3: A Sparse Linear Equation Solver, available at http://www.netlib.org/sparse/

[7] J. Lee, K. Kim, Y. Huh, P. Bendix and S. Kang. Chip-level charged-device modeling and simulation in CMOS integrated circuits. *IEEE Trans. Computer-Aided Design*, vol. 22, no. 1, pp. 67-81, 2003.

[8] Q. Li, Y. Huh, J. Chen, P. Bendix and S. Kang. Full chip ESD design rule checking. *Proc. ISCAS*, pp. 503-506, 2001.

[9] T. Li, S. Ramaswamy, E. Rosenbaum and S. Kang. Circuit-level simulation and layout optimization for deep submicron EOS/ESD output protection device. *Proc. IEEE Custom Integrated Circuits Conf.*, pp. 159-162, 1997.

[10] T. Li, C. Tsai, E. Rosenbaum and S. Kang. Modeling, extraction and simulation of CMOS I/O circuits under ESD stress. *Proc. ISCAS*, vol. 6, pp. 389-392, 1998.

[11] O. J. McAteer. *Electrostatic Discharge Control.* McGraw-Hill, New York, NY, 1990.

[12] H. Qian, S. R. Nassif, and S. S. Sapatnekar. Random walks in a supply network. *Proc. DAC*, pp. 93-98, 2003.

[13] H. Qian and S. S. Sapatnekar. Hierarchical random-walk algorithms for power grid analysis. *Proc. ASPDAC*, pp. 499-504, 2004.

[14] N. Sclater. *Electrostatic Discharge Protection for Electronics.* TAB Books, Blue Ridge Summit, PA, 1990.

[15] S. Sinha, H. Swaminathan, G. Kadamati and C. Duvvury. An automated tool for detecting ESD design errors. *Proc. EOS/ESD*, pp. 208-217, 1998.

[16] M. Zhao, R. V. Panda, S. S. Sapatnekar and D. Blaauw. Hierarchical analysis of power distribution networks. *IEEE Trans. Computer-Aided Design*, vol. 21, no. 2, pp. 159-168, 2002.