Automatic Layout Based Verification of Electrostatic Discharge Paths

Paul Ngan (1), Robert Gramacy (2), Ching-Kwok Wong (3), Dan Oliver (4), Theo Smedes (5)

(1) Philips Semiconductors, 811 E. Arques Ave., M/S-35, Sunnyvale, CA. 94088-3409, USA
tel.: 408-991-4992, fax: 408-991-2566, e-mail: paul.ngan@philips.com

(2) University of California, Santa Cruz, (3) Kestrel Solutions

(4) Philips Semiconductors, Sunnyvale, (5) Philips Semiconductors, Nijmegen

Abstract - This work describes an ESD path verification methodology based on layout parasitic extraction. This approach was implemented in Cadence DFII. It provides information about the preferred ESD path between two pads and estimates the peak pad to pad voltage. The path can also be overlaid on the layout view. The methodology was applied to a 0.5um BiCMOS design to improve its ESD robustness. In that case, weak ESD paths overlooked during conventional design reviews were identified and corrected. The ESD robustness improved from 1.0kV to 2.5kV and 100V to 250V for HBM and MM respectively.

I. Introduction

Verification of Electrostatic Discharge (ESD) robustness of today's VLSI designs is becoming more and more difficult and time consuming. In many companies, ESD experts, together with designers/product engineers still primarily perform this function right before the tape-out. Depending on the expertise of the group and complexity of the design, the adequacy of the protection is often not fully understood until the "silicon" characterization phase. The result is design iteration – incremental improvements in ESD protection based on the characterization. Unfortunately, this is often in conflict with tight time-to-market requirements. Therefore, the ESD verification methodology has to be an integral part of the design flow, rather than a singular activity finished before tape-out. Automation is needed to tackle large designs. In addition, the verification methodology should be able to intuitively "pin-point" the weaknesses of the design and allow designers to "experiment" with possible fixes.

This paper presents an ESD path verification methodology developed to improve product ESD reliability as well as ESD robustness success rate for "1st silicon".

The paper is organized as follows. In Section II, the possible solutions to the ESD verification are presented with a brief description of pros and cons. In Section III, the chosen methodology will be described in detail. A description of the implementation and some representative examples follow in Section IV. In Section V, a case study of one of 0.5um BiCMOS products is presented. Section VI is a brief discussion of the strength of the methodology and possible extensions. In Section VII, conclusions for this work are given.

II. Possible Solutions

A good ESD verification tool for today's VLSI design is essential. However, there are many factors involved that could affect the final ESD robustness. Such factors include generic protection network topology, device layout sensitivity and path parasitics. In general, ESD verification can fall into two categories: 1) Design Rule Check (DRC) and 2) Robustness Check.

DRC is used mainly to prevent layout errors. It is set up to check for situations like the familiar contact to polysilicon spacing rule. Ref [1] describes one such approach. This approach is layout based but often does not provide adequate information about ESD paths. In addition, since the rule is normally set up for one particular technology, a new set of rules might have to be set up for each technology. However, the implementation is relatively straightforward if all the rules are in place.

On the other hand, the Robustness Check is used to find the preferred path of the ESD event. Such a

methodology will identify the path which the ESD current is likely to choose and will also give electrical details about this path. It can be schematic-based [2] as well as layout-based. Both approaches can identify ESD weakness inherent in a design.

The schematic based approach is attractive because designers can easily simulate different responses for different protection topologies. However, it relies heavily on the availability of appropriate compact device models, especially in the high current regime. (For example, R_d of a diode at 1A is not the same as at 100μA.) In addition, the layout parasitics are often excluded which could be a major factor in determining the preferred ESD path. In other words, the predicted strength of the probable path will be inaccurate, or worse, the simulated paths might not be the actual ones.

The layout-based approach has an obvious advantage in that parasitics and path resistance can be easily included. However, this requires extracting parasitic path resistance from the layout and then referencing back to the connection in the schematic. Similar to the previous approach, it relies on accurate circuit simulation to analyze the possible paths [6, 7].

This paper addresses a different approach which is described in detail in next section. It is layout based and the effects of parasitics are included. Preferred ESD current paths can still be identified but circuit simulation is not required and there is no need for accurate high current device models.

III. The Chosen Solution

The methodology is designed to address two fundamental questions for checking ESD robustness:

1. Does a (preferred) ESD path exist? If so,
2. What would be the pad voltage at peak current?

Our solution for checking ESD robustness is therefore, layout-based and has the following features:

1. It will identify a preferred (lowest resistance) ESD path and provide an estimate of the pad voltage.
2. User Input / Output interfaces are intuitive.
3. It can provide a list of estimated clamp voltages for all pin combinations.
4. It is technology independent. That means, it can be implemented into a different technology library with minimal reprogramming.
5. The extraction is done at cell level. Therefore, detail compact device models are not needed. Instead, Transmission Line Pulsing (TLP) characterization data is used for each cell element (diodes and supply clamps) to estimate the pad voltage.

IV. Implementation

The tool was implemented in Cadence DFII version 4.4. The basic operation of the tool can be summarized with flowchart as shown in Figure 1.

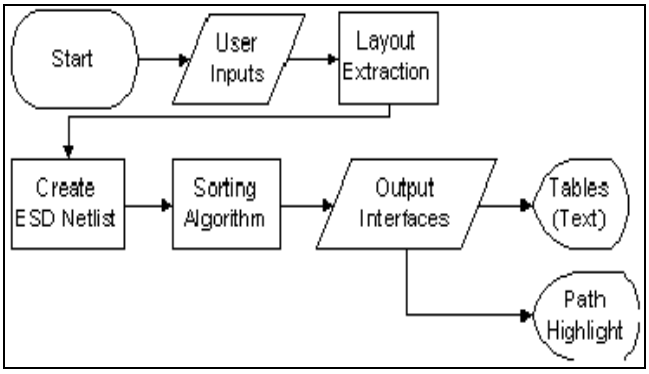

Figure1. Analysis Flow of the methodology

Once the tool is invoked, the user can change the default settings in the preference window (Figure 2.)

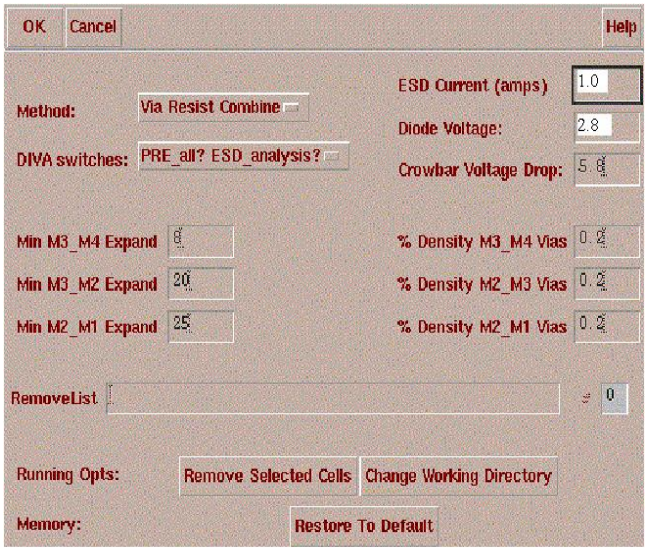

Figure 2. Initial Input menu. "DIVA" is name of the layout extraction tool. "Crowbar" is the generic name used for the supply clamp element.

The most important inputs are ESD current, diode voltage and the supply clamp element voltage at the selected ESD current. Both diode and supply clamp voltage should be obtained from TLP characterization. For example, in Figure 3, a typical TLP curve for a diode element is shown. The diode voltage can be read directly from the graph (~2.8V @ 1A).

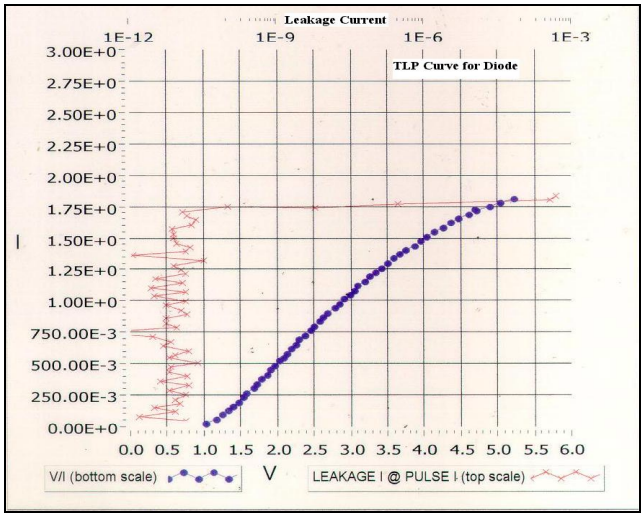

Figure 3. A typical TLP curve for diode.

The Parasitic Extraction Routine will then extract all the metal resistances (including via resistances) for the layout. The basic elements (bond pad, diodes, etc.) are maintained at the cell level and are not extracted. All the cell elements will then be recombined to form a netlist (ESD netlist). Once this is done, the ESD path (between two pads) is determined based on the Dijkstra shortest path algorithm [3]. All conducting paths are considered separately and the pad voltage is calculated by the following formula:

$$V_{pad} = V_{clamp} + \sum_i V_{diode,i} + I_{ESD} * \sum_i R_i$$

where R_i is the parasitic resistance elements between two pads along the path. The chosen path is the one with lowest pad voltage. The path and pad voltage can be listed in text format as shown in Figure 4. A complete table listing all pin combinations can also be generated. Furthermore, the path can be superimposed on the layout. In Figure 5 and 6, the ESD path is highlighted by rectangles.

```
from [X|BPESDSH_1~Pad], to:
                    X|BPESDSH_4~Pad      bgt   11.3673v  ==>
X|BPESDSH_1~Pad -(DD1)-> X|BPESDSH_1~Pwr -(X|BPESDSH_1) -->
~450 -(R+382) [resistor element] --> {similar resistor elements} -->
X|BPESDSH_4~Sub -(DD2)-> X|BPESDSH_4~Pad
```

Figure 4. Typical Output List, DD1 and DD2 are the diode elements. X|BPESDSH is the name of the object, bond pad in this case. The intermediate resistor elements (metal) have been omitted for clarity.

The runtime for the Dijkstra algorithm is proportional to the number of pads. Typical runtime for a 20 pad padring cell, similar to the one described in Section V, is < 1 minute. For larger design (i.e. more pads), the run time increases roughly quadratically.

The Dijkstra sorting algorithm actually generates the upper bound estimate for the pad voltage since it only picks out the lowest resistance path. A separate algorithm called Convex Flow Analysis can take all possible forward current paths into account and thus provides a more realistic estimate of the peak voltage [4]. A brief discussion about the Dijkstra algorithm and Convex Flow analysis is provided in Appendix.

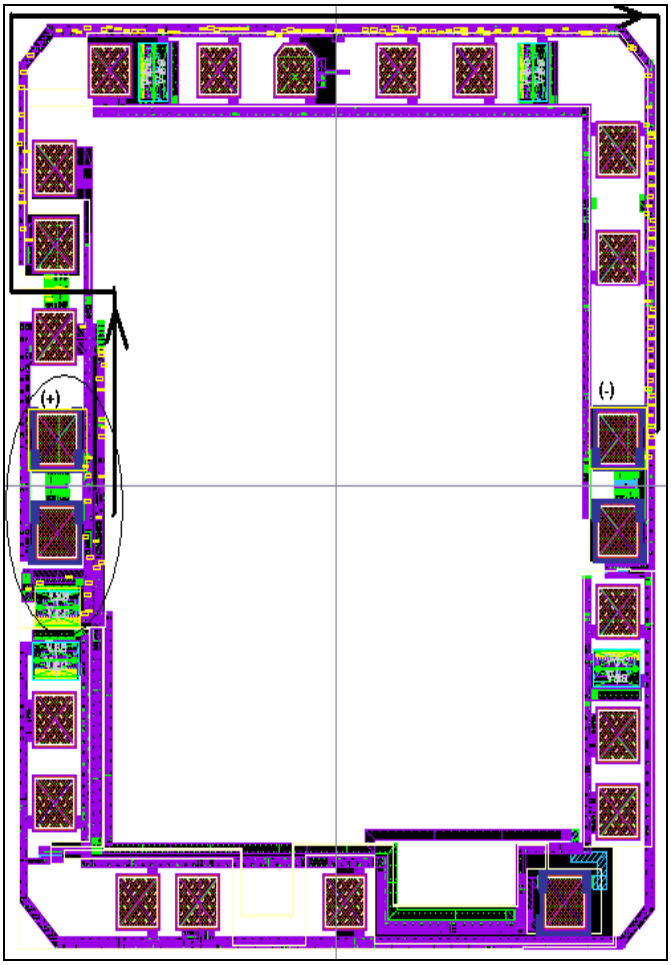

Figure 5. Path highlight function, the path is highlighted by rectangles. The black line shows the "flow" of the rectangles.

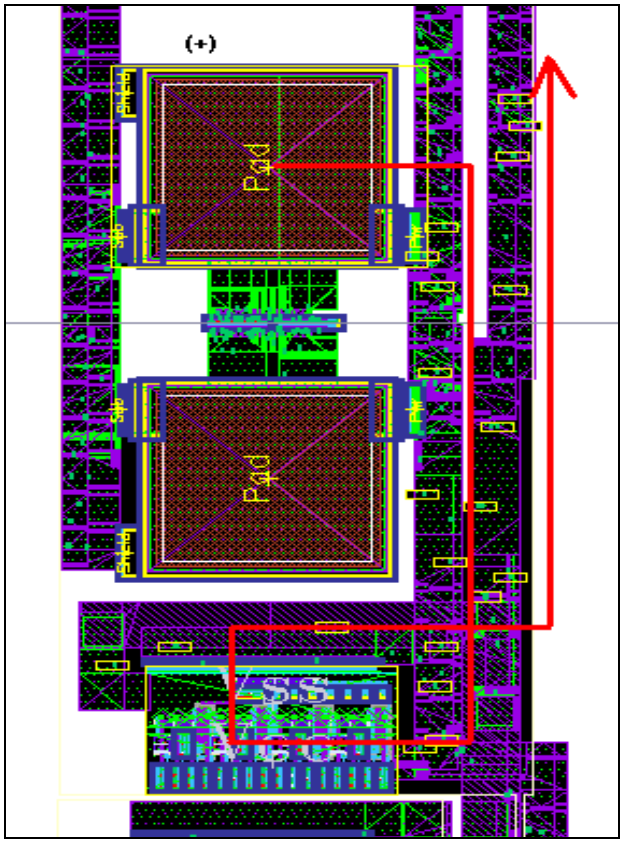

Figure 6. Close up of encircled area in Figure 5. The arrow shows the flow of the rectangles.

V. Case Study Results

This methodology described above was applied to a product in 0.5um BiCMOS technology. The protection network was based on standard diode plus supply clamp network. The initial ESD characterization showed the device had only ~1kV HBM and 100V MM threshold. Detail characterization and Failure Analysis indicated a diode to positive supply was damaged (Figure 7).

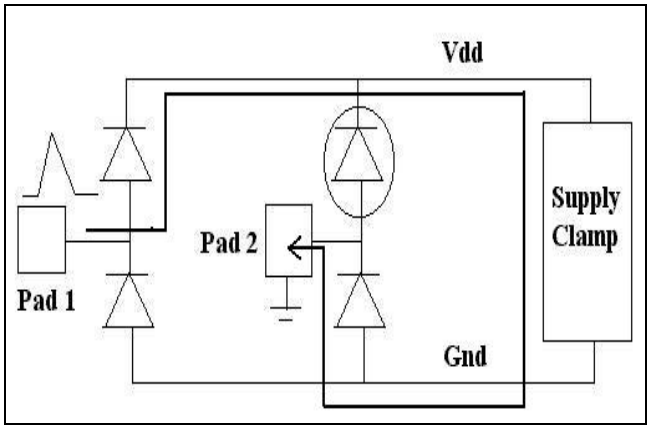

Figure 7. I/O pair that shows weak ESD robustness. The ESD pulse is positive from Pad_1 to Pad_2. The damaged diode is circled. The arrow is the expected direction of the ESD current.

Applying the new methodology, it was determined that the pad voltage was excessive at the peak ESD current between pairs of I/O pins. In fact the voltage between the Vdd and Pad_2 in Figure 7 exceeds the diode breakdown voltage (~16V). The primary reasons are the high turn-on voltage (V_{t1}~13V) of the supply clamp (bipolar latchback device) and non-optimized metal routing (high metal bus resistances). Some paths have total metal bus resistance as high as 17Ω. The ESD protection network was therefore ineffective. Based on this information, the supply clamp was changed to an active triggered crowbar device (redesigned with compacted layout based on the original work by Dr. R. Meyer [5]) with V_{on}~5V and the metal bus routing was also optimized (Figure 8 and 9). The predicted pad voltage were dramatically reduced after layout optimization (~3x reduction in some cases). Table 1 shows the results for an I/O pair as shown in Figure 7.

Table 1. Results of the estimated peak Pad_1 voltage @ 1A.

Condition	Est. peak Pad_1 Voltage @ 1A
Original topology (Figure 8)	30V
New topology (Figure 9) – Dijkstra algorithm	14V
New topology (Figure 9) – Convex Flow algorithm	11V

The voltage was thus well below the diode breakdown voltage. The subsequent ESD characterization showed the ESD robustness was improved to 2.5kV HBM and 250V MM.

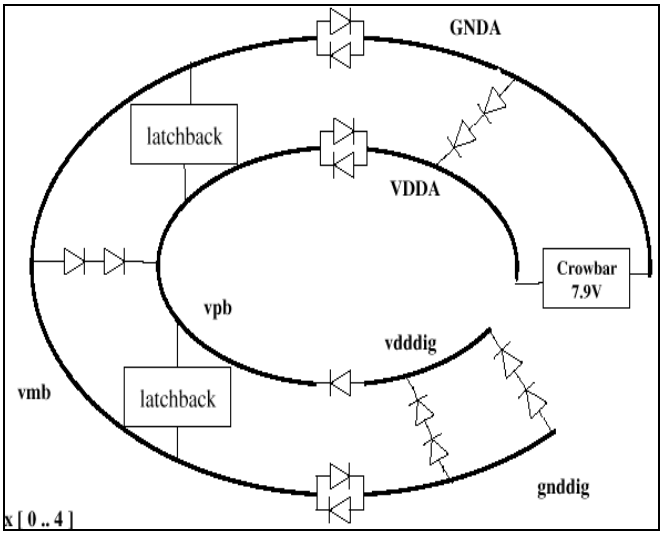

Figure 8. Initial ESD Protection Topology. (HBM threshold ~ 1kV)

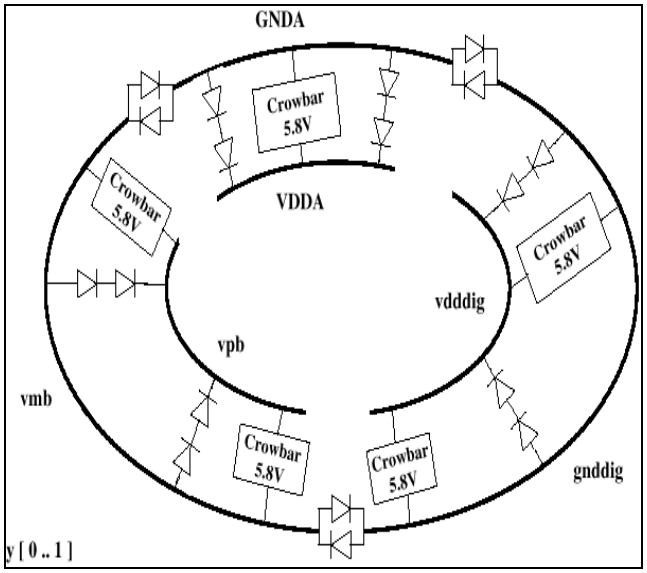

Figure 9. Protection Network after Optimization. (HBM threshold > 2.5kV).

VI. Discussion

The main advantage of the methodology is:

1. It is intuitive. It allows the designers to check out the layout deficiencies quickly with quantitative information.
2. Since there is no circuit simulation requirement, detailed compact device model is therefore not needed.
3. The separation of extraction and sort algorithm allows flexibility of adapting this approach to different technology with minimal changes.

The use of actual TLP characterization data avoids detailed compact device model development. However, the transient behavior cannot be captured in the path calculation. As an alternative, designers can run the simulations at several current levels and check the changes in pad voltages. This methodology can potentially be improved by combining with a schematic based approach (with high current device models) to allow more accurate simulation. It can also be used in conjunction with the Design Rule Check to provide complementary information for design and layout engineers.

VII. Conclusions

A new layout based methodology in verifying ESD robustness has been described. The approach can allow designers to check the adequacy of the ESD protection network quickly as part of the design flow with quantitative information. Circuit simulation is not required and detailed compact device model is not needed. Due to the separation between parasitic extraction and sorting algorithm, this approach can be extended to other technologies with minimal changes. In addition, this approach can potentially be combined with a schematic based approach (with high current device models) to allow more accurate simulation. It can be used together with the Design Rule Check to provide complementary information. Finally, this methodology has been successfully applied to improve one of the product ESD performance, from ~1kV to 2.5kV HBM.

VIII. Acknowledgments

The authors would like to thank the technical support from the Philips Semiconductors' CAD Engineering and Business Line Cellular America's Design Engineering. Discussions and encouragement from different engineering teams within Philips Semiconductors are also greatly appreciated.

IX. References

1. S. Sinha, H. Swaminathan, G. Kadamati and C. Duvvury, "An Automated Tool for Detecting ESD Design Errors", EOS/ESD Symposium, 1998, p. 208
2. M. Baird, R. Ida, "Verify ESD: A Tool for Efficient Circuit Level ESD Simulations of Mixed Signal ICs", EOS/ESD Symposium, 2000, p. 465.
3. E. Dijkstra, A note on two problems in connexion with graphs: Numeriche Mathematics 1, 1959, p.269-271
4. Robert Gramacy, "Shortest Paths and Network Flow Algorithms for Electrostatic Discharge Analysis", Senior Thesis, U.C. Santa Cruz, 2001.
5. R. Meyer et. al., "New ESD Protection Schemes for BiCMOS Process with Application to Cellular Radio Designs", ISCAS 1992, p. 2699-2702.
6. T. Li, C.H. Tsai, E. Rosenbaum, S. Kang, "Modeling, Extraction and Simulation of CMOS I/O Circuits under ESD Stress", IEEE Int. Symposim on Circuits and Systems, 1998, pp. 389-392.
7. T. Li, S. Kang, "Layout Extraction and Verification Methodology for CMOS I/O Circuits", Design Automation Conference, 1998, pp. 291-296.

X. Appendix

This appendix contains a brief description of the Dijkstra and Convex Flow algorithm. Both algorithms attempt to minimized a given "cost function" *C*.

1. Dijkstra algorithm: This algorithm, based on the work of Dutch mathematician E. Dijkstra, involves the creation of a shortest paths tree from the source *S* to all sinks *T* by intermediate possible nodes one at a time. It can be shown that if the path from *S* to an intermediate node *V* is itself a shortest path, and the path from *V* to *T* is also a shortest path, then *S* to *T* must also be the shortest.

Figure 10 shows an example of paths from V_1 to V_k through some intermediate nodes. Each node pair (V_i,V_{i+1}) has an associated cost function $c(V_i,V_{i+1})$ which is the resistance value of the path element. The algorithm seeks to determine the path which yields the minimum value for $\sum_{i=1}^{k-1} c(v_i, v_{i+1})$.

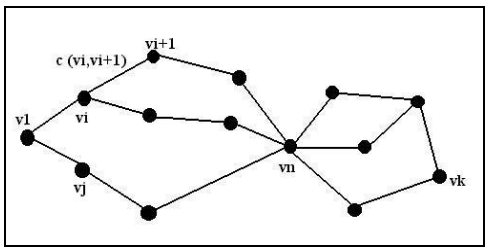

Figure 10. An example of paths from V1 to Vk with intermediate nodes is shown. Each node pair has an associated cost function $c(v_i, v_{i+1})$

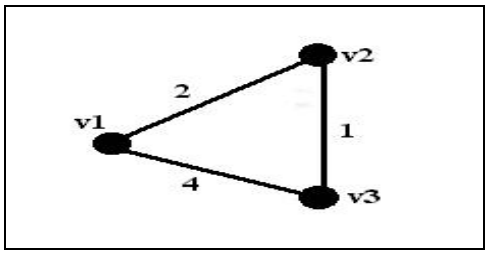

Figure 11. An example of two possible paths from V_1 to V_3 is shown. Each node pair has an associated cost function with value shown next to the arc

To illustrate the process, let's consider two possible paths between V_1 and V_3 as shown in Figure 11. The process starts from V_1 and the total cost function for each path is calculated. The process repeats until all paths between V_1 and V_3 are considered. The two paths are V_1/V_3 and $V_1/V_2/V_3$. The cost functions are then compared and the shortest path is then determined. The result will be the lowest resistance path from V_1 to V_3. In this example, the shortest path is $V_1/V_2/V_3$ and its cost function equals 3.

2. Convex Flow Algorithm: Although the Dijkstra algorithm can determine the lowest resistance path between two points quite efficiently, it can be shown that the estimated pad voltage will be the upper bound or worst case value. It is because each conducting path is considered separately. Convex Minimum Cost Maximum Flow (or Convex Flow) algorithm is implemented to solve this problem. Similar to Dijkstra algorithm, we consider all path(s) from source *S* to sink *T* and all the intermediate nodes. The system constraint is the total amount of available current.

This algorithm can be mathematically put into a form similar to an optimization problem using quadratic programming technique. The algorithm determines the equilibrium current for each arc by minimizing cost function $\sum x_{ij}^{\ 2} \cdot c_{ij}$ where x_{ij} is the flow (fraction of total current) and c_{ij} is the cost function (resistance) between node (i , j). No paths will be ignored. The only difference among paths is the amount of flow (current).

The tradeoff, compared to Dijkstra algorithm, is that the computational time for Convex Flow algorithm is considerably longer.

To illustrate the difference between the two algorithms, let's consider a test case of a resistor cube as shown in Figure 12 [4].

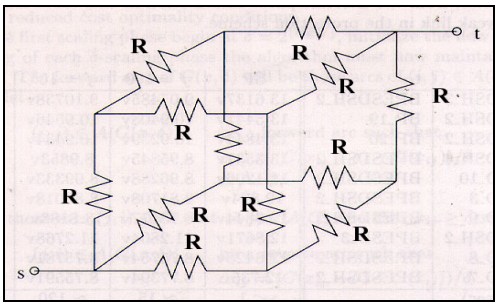

Figure 12. A resistor cube between two nodes S and T. Each resistor has equal resistance value R.

Consider a resistor cube between two points, *S* and *T*. Each edge has a resistor value R. There are altogether 12 resistors. From basic electrical network theory, we can deduce that the effective resistance $R_{eff} = (5/6)*R$. The results for Dijkstra and Convex Flow algorithm are shown in Table 2.

Table 2. Comparison for Resistor Cube

	Ideal value	Dijkstra	Convex Flow
R	5/6*R	3R	0.8332*R
Time (s)	---	<< 0.5	~2

Dijkstra algorithm picked up the three resistors along (one of) the three edges, as expected. The Convex Flow algorithm produced relatively accurate estimate of the equivalent resistance. The computational time for Convex Flow algorithm, however, is 4-5 times longer.