Static Verification of Test Vectors for IR Drop Failure

Aman A. Kokrady and C.P. Ravikumar
ASIC Product Development Center
Texas Instruments India
Bangalore, India 560017
{koko,ravikumar}@india.ti.com

Abstract

ATPG tools generate test vectors assuming *zero delay model* for logic gates. In reality, however, gates have finite rise and fall delays that are dependent on process, voltage, and temperature variations across different dies on a wafer and within a die. A test engineer must verify the vectors for timing correctness before they are handed off to the product engineer. Currently, validation of tests is done using dynamic simulation of the circuit using the test vectors. A test vector is invalidated if it cannot reliably distinguish between a good and a faulty circuit under the signal placement and observation error window of the tester equipment. Since structural tests can result in much more switching activity in the circuit than what is estimated during normal functioning, the IR drop in the power & ground lines can be significant, adversely impacting path delays. As a result, the validation performed by simulation can be error prone. Oversizing the power rails to address this problem impacts the yield. We therefore propose the verification of test vectors for IR drop failure and present a flow for identifying failing vectors. Attempting to address this verification in dynamic simulation will force the use of circuit simulation or mixed-level simulation techniques, which are expensive in terms of run time. We discuss a static approach to validate the test vectors for failure in the presence of IR drop problems.

1 Introduction

In order to simplify the process of test vector generation, an ATPG tool assumes that logic gates and wires are ideal components with no parasitic delays. Real gates have delays that depend on the manufacturing process used (strong or weak) as well as operating conditions such as temperature and voltage. The operating conditions can vary from one die to another, and within the die. A system-on-chip design built using sub-micron technology exhibits several nonlinear effects such as crosstalk coupling between wires, resistive drop in the power supply lines, dynamic voltage drop due to power line inductance, and so on. A test vector becomes invalid if it cannot reliably distinguish between a good and a faulty circuit under the signal placement and observation error window of the tester equipment. After a test vector is applied, the output of a gate may glitch one or more times before settling down to the final value. The presence or absence of glitches and the settling delay can vary from one instance of the circuit to another due to variations in Process-Voltage-Temperature (PVT). Due to these problems, test vectors are validated for timing stability through simulation at several PVT corners.

Unfortunately, simulation is a CPU-intensive task since the circuits are large (several million gates), the number of test vectors is large (several million vectors) and the number of PVT corners is large. This is despite the fact that logic simulators do not comprehend delay variations due to deep submicron effects such as crosstalk and IR drop. The quality check made on test vectors, therefore, is suspect if the impact of second order effects is substantial. Normally, a designer builds extra margin into the design to overcome these problems; e.g. the power lines can be made wider to reduce the impact of IR drop.

At-speed tests [3] are at largest risk due to IR-drop related tester failure. In this paper, our objective is to describe a static approach that can factor in the impact of IR-drop during the validation of transition delay fault test vectors. We believe ours is the first attempt to solve this problem. Since we avoid simulation of netlists, our solution is much faster. Essentially, our approach is to eliminate a large number of vectors based on a short-listing algorithm that estimates the switching activity on power rails for each vector and rejects vectors that result in more switching activity than a rail can tolerate. This algorithm takes into account the strengths of power rails. The vectors so short-listed are analyzed for toggle count and the resulting IR-drops are analyzed using an IR drop estimation tool. The paper is organized as follows. In the following section, we briefly introduce the background required for the paper. Section 3 explains the proposed flow for validation of transition delay tests and algorithms used for estimating the power rail switching. Implementation of the flow is explained in Section 4. Results are included in Section 5. Conclusions are presented in Section 6.

2 Background

At-speed testing verifies the timing correctness of the manufactured circuit. This form of testing is critical for nanometer technologies since their timing is impacted due to a variety of reasons such as crosstalk and IR drop. Embedded memories are almost always tested at-speed using built-in self-test. Apart from functional at-speed tests, path delay and transition delay tests are two forms of structured at-speed tests for logic [2,3]. In the former, we select a set of paths whose timing is critical to the correct at-speed functioning of the chip and ensure that the effect of rising and a falling transitions at the input pin of a path can be observed at the output pin of the path. Path delay testing is expensive since the test data volume for this form of testing is high even for modest coverage. In transition delay testing (also called gate delay testing), the fault model consists of a slow-to-rise transition or a slow-to-fall transition at the output of a gate. The number of transition delay faults is linear in the number of gates, as opposed to the number of path delay faults, which is exponential in the number of gates.

The test for a transition delay fault in a full-scan logic circuit consists of the following steps, assuming that the fault is a slow-to-rise transition at the output of a gate *g*. See Figure 1.

ICCAD'03, November 11-13, 2003, San Jose, California, USA.
Copyright 2003 ACM 1-58113-762-1/03/0011 ...$5.00.

1. Find a scan test vector V_1 to place a logic 0 on the output of *g*. V_1 may be applied at a slower speed to avoid excessive dynamic power dissipation due to switching activity that will result from scanning in V_1.
2. Find a way to launch a rising transition at *g*. Suppose this can be done by applying a test vector V_2. In the "launch by capture" method of ATPG, we generate V_1 such that V_2 is the state of the circuit after applying vector V_1 and then switching to normal mode of operation. Thus, after scanning in V1, we deactivate Scan Enable signal and do a capture of the circuit state at a slow speed. The transition has now been launched.
3. We now do a "fast capture" and capture the state information in the scan flops after time T, which is the period of the device under test. If a delay fault exists in the circuit, the effect of the delay fault will be captured into the scan flops.
4. Scan out the contents of the scan flops.

If the switching activity causes considerable current to flow in the power line, the resistive drop (as well as the L di/dt drop) in the power line can lower the power supply to the cells. Since the cell delays are inversely proportional to the power supply, the responses of the circuit can be delayed more than the expected value and fast capture will fail to capture the correct response. This will result in a potential yield loss, since the circuit is indeed functional, but the test declares it faulty. The same problem holds for memory BIST tests. Memories are tested at-speed using built-in self-test circuitry. When designing memory BIST circuitry, the designer often concentrates on minimizing the number of controllers and test application time. In the process, a large number of memories may get tested at the same time; such memory access patterns may not occur in functional mode. This can again cause IR-drop induced speed failure of memory BIST vectors.

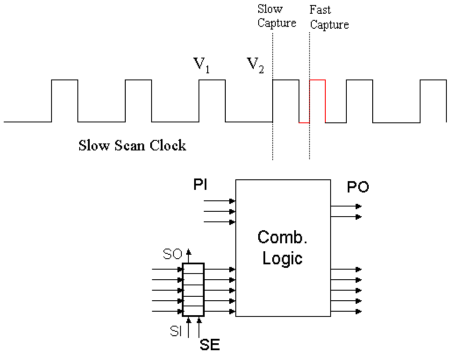

Figure 1: Fast Capture in Transition Delay Test

One way to overcome the above problem is to simulate the test vectors taking second order effects into account. However, since there are a large number of test vectors, such a solution is very expensive. We propose a static method to verify test vectors. As part of our validation flow, we are interested in identifying test vectors that sink large amounts of dynamic power. The subject of power estimation and impact of instantaneous power on power and ground busses has been reported by several researchers. Before we survey the pertinent literature, we comment that our work is different in the following aspects: (1) Much of the existing work on current estimation in power and ground lines is for combinational circuits, whereas we consider full scan circuits including embedded memories. (2) We report results on industrial-sized benchmarks, whereas the existing literature focuses on small benchmark circuits with less than 10,000 gates. (3) Much of the recent work on power and ground current estimation focuses on second order effects such as simultaneous switching noise, Ldi/dt effect, primary input misalignment, etc., and the techniques reported provide accurate estimations on small circuits. In this work, our focus is on IR drop without considering the second order effects, since modeling them would impact the run times on industry-sized benchmarks adversely without adding significant value. An early power estimator was reported by Haroun et al [7]. Commercial power estimation tools are also now available, e.g. [13]. A pattern-independent maximum current estimator was reported by Kriplani et al [9], whose *iMax* algorithm can report the current waveform at every contact point. Bai et al [1] extended this technique to estimate the worst-case voltage drop in digital circuits; they reported results on a 4 × 4 combinational multiplier. Kriplani et al. provided improved upper bounds on power supply current by resolving the signal correlations in the circuit [8]. Dutta et al [6] considered automatic resizing of power and ground busses. Chaudhry et al presented a "compaction" technique to construct a set of vectors that maximize the current in a combinational circuit [4]. Krstic et al also presented a method for vector generation to maximize power supply current in combinational circuits [10]. A hybrid methodology for switching activity estimation, which considers both simulation and probabilistic estimation techniques is presented by Cheng et al [5]. They used simulation for control paths and probabilistic techniques in data paths. In the existing approaches towards vector construction for worst-case power, the actual functionality of the circuit is ignored. If we have several combinational blocks separated by (scan) registers, an extension of the above techniques for constructing a worst-case vector may not be possible for two reasons: (1) Two vectors which result in worst-case power dissipation in two separate blocks A and B may have contradicting inputs. (2) Initialization of scan chains to simultaneously maximize the power in two or more blocks may be impossible.

In this paper, we attempt the solution of a somewhat different problem, where structural test vectors are provided and they must be verified for potential failure on the tester due to IR-drop and resulting delay faults.

3 IR-Drop Aware Validation of Test Vectors

Since a SoC has several million gates and several megabytes of memory, it becomes necessary to have multiple power rails in order to supply power to all the circuits. The power rails are normally optimized taking the functional power dissipation into account. Let us first look at the validation of delay test vectors assuming full-scan based test application. Generally, the natural hierarchy in the design is used in designing the scan chains for the chip. Similarly, a hierarchical approach is followed during power supply design. It is possible that supply rail *j* is connected to several scan chains. See Figure 2 below, where we show two sub-blocks B_1 and B_2 in a hierarchical design. We show a scan multiplexer that allows sharing of the scan input and scan output pins between the two blocks. The blocks are tested one at a time. Although a single scan chain is shown (as a dotted line) we can have several scan chains in practice. The figure shows two power rails. Rail R_1 supplies power to gates in both blocks, whereas rail R_2 supplies power to only block B_2.

Validation of a test vector for IR-drop failure must test for the following conditions:

C1: During test application, IR drop in at least one power rail will exceed the margin built into the design by the physical designer.

C2: There exists a path that is exercised by the test vector which is adversely impacted by the IR drop.

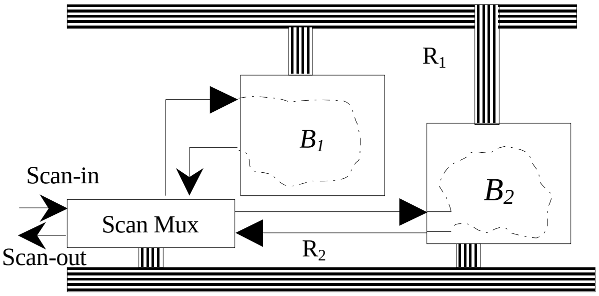

Figure 2: Power Rails and Scan Chains

Condition C2 is the stronger condition, and testing for its application will be time consuming. We therefore propose a static method to shortlist test vectors based on condition C1.

3.1 Overview of the method

Our IR-drop aware test vector validation algorithm is called TestRail. The primary inputs include transition delay test vectors the gate-level netlist, the physical layout database, and the delay information for the circuit in Standard Delay Format (SDF).

Vector Short-listing

From the information in test vectors, we extract the toggle count on each of the primary inputs, outputs, and scan flops of the circuit. Let T be the set of test vectors and N be the set of primary inputs, primary outputs, and scan flops in the circuit. For each $t \in$ T and for each element $n \in$ N, let TC_{tn} be the toggle count on n due to application of t. A naïve method to validate a test is to consider the sum $\Sigma\, TC_{tn}$ over all n and apply a threshold. This method does not factor in the power rail information. An improved technique is to consider the mapping of flops to power rails; let R be the set of power rails and let $M(n,R)$ be defined to be a 0/1 variable which is 1 if and only if element n is powered by rail R. Further, let $\sigma(R)$ indicate the strength of the rail R. We choose the strength function to return a positive integer that is in inverse relation to the threshold on the amount of toggling activity the rail can permit. We propose the use of the following measure to validate a test:

$$\mathrm{A}(t) = \Sigma_{\mathrm{R}}\, \Sigma_{\mathrm{n}}\, TC(t,n)\, .\, M(n,R)\, .\, \sigma(R)$$

The reader may verify that the activity factor A(t) reflects both the toggling activity generated by test t as well as the stress this causes to the power rails. The vector short-listing algorithm we propose evaluates A(t) for all test vectors and constructs a histogram of A(t). The vectors whose activity factor is far above the average (larger than average + k. standard deviation, where k can be specified) are short-listed. The subset of test vectors short-listed is subjected to further analysis, namely, power rail analysis and critical path analysis.

Power Rail Analysis

A switching activity propagator propagates the toggle information from the set N to internal nodes of the circuit [15]. An IR drop estimator is used to obtain estimates of IR drop in power rails; we used a commercial software (Synopsys Astrorail [14]). The tool can generate a color-coded plot which displays in red the areas in the chip where the IR drop exceeds the margin. Typically, such areas are in central portions of the chip. We use the short-listed vectors to compute the toggling on the nets and map this information to IR drops in the AstroRail flow. A vector for which IR drop violations are pointed out by the power rail analysis step is treated as a candidate for further analysis.

Critical Path Analysis

In this step, we wish to identify vectors which activate critical paths that may fail due to the adverse impact of IR drop on gate delays. A static timing analysis tool (PrimeTime [12]) is used to compute the criticial paths in the circuit prior to IR drop analysis. Three different methods were considered for updating the chip delay information after worst-case IR drop has been computed for all nets. Device simulation through SPICE will be an accurate approach, but impractical due to large design sizes. In the non-Linear data model (NLDM) approach, slew values are represented for different combinations of power supply voltage, temperature and process parameter. A query on this database is used to compute the change in cell delay. In the scalable polynomial delay model (SPDM) approach, the cell delay is represented as an equation in terms of process, voltage, and temperature parameters. This approach is fast and less memory intensive. In the interest of run-times, we actually used a simple linear model for updating the cell delays. The cell delay is updated according to the equation

$$\text{Delay}' = \text{Delay} + \alpha.\ \text{IR}/\text{V}_{\text{DD}}$$

where IR is the estimated IR drop in the supply voltage to the cell and α is a library-specific positive real constant which we tuned through experimentation. We annotated the SDF file with the new delays computed based on the equation above.

TestRail then performs a static timing analysis on the circuit using the updated SDF file. All the critical paths are reanalyzed for any setup, hold or slack violation. If no violations are reported then TestRail reports that the vectors will not fail on the tester due to IR drop problems.

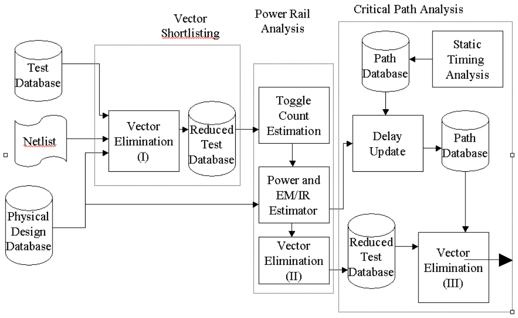

Figure 3: TestRail Software Flow

4 Implementation

We implemented TestRail in about 4500 lines of Perl code and about 1000 lines of C code. The organization of the TestRail flow is depicted in Figure 3. The three phases of the TestRail flow are marked in dotted boxes. The figure shows the three stages of vector filtering. The transition delay test vectors database is given as input to TestRail. The Verilog gate-level netlist format and the physical design database is provided in Avanti Milkyway Database format. Several in-house and commercial tools were used in the implementation. A C program, along with an in-house tool, is used to parse the test vector file and write the "Value Change Dump" (VCD) file for the scan flops in the design at the fast capture instants. Dumping only the select information reduces the memory and time complexity of all the tools in the flow. Another in-house tool was used to propagate toggle count information into the internal nodes of the circuit. Synopsys AstroRail [14] is used to estimate the average IR drop occurring on the nets. The tool uses the toggle count information to generate the estimates of voltage drops. The AstroRail output file is filtered for violating cells depending upon a threshold IR drop. The threshold is a value above which the voltage drop may cause a delay problem. Synopsys PrimeTime [12] is used to perform static timing analysis.

5 Results

We tested the flow presented in the previous sections on two industrial designs, which we will call Chip-A and Chip-B (not the actual names). Chip-A is a memory-intensive design with 70K logic gates and 64KB memory, with a single scan chain (flip-flops). This chip has only two power rails, namely, VDD and VSS. Chip-B has around 3 million gates and about 840KB of read/write memory, and has 7 scan chains (flip-flops). The chip has 12 power rails. The design Chip-A was selected as a pilot test case to verify our flow. In this relatively small design with a single scan chain and a single VDD power bus, it was possible for us to manually construct test vectors that will fail due to IR drop. We also constructed functional test vectors for this circuit. There are 5 repairable RAMs in the design, all powered by the same bus, and all tested concurrently using SMARCHKBCil algorithm (serial March, checker board) during test mode. In addition to the memories, the design also has a 32-bit sequential multiplier, which was also tested using Scan ATPG transition-delay test vectors generated using TetraMAX. The test vectors were taken through both the TestRail flow as well as the standard simulation-based verification flow. The test vectors passed the latter flow, but the TestRail flow caught the failing vector, as shown in the AstroRail plot of IR-drops (Figure 5(a)). The central portion in the chip is red since IR drop here exceeds the margin. The failing vector exercised all the five memories using BIST *and* the multiplier using the transition delay test. To contrast, we also created functional test vectors which exercised only subsets of the five memories (it is not functionally possible to exercise all memories at the same time). The TestRail flow did not find any problems with such functional vectors (see Figure 5(b)). As was pointed in the previous sections, the IR-drops exceeding the margin is a necessary but not sufficient test for the failure of the test vector. We further verified the test vectors that failed the margin test by taking it through the static timing analysis flow and found that the vectors indeed failed due to timing faults in the critical path.

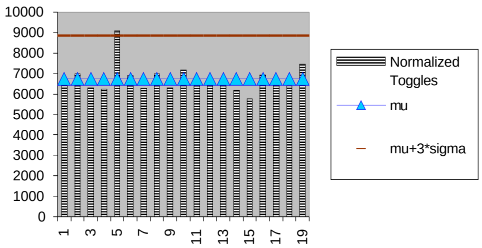

Figure 4: Toggle Counts for Chip-B for 19 vectors

Figure 4 shows the results on the vector short-listing technique proposed in this paper. We plot the toggle counts for the initial 19 test vectors (selected randomly to illustrate the point). As we can see, the toggle count for Vector 5 is unusually large, larger than the average+3*standard deviation, making it a strong candidate for shortlisting. In fact, Vector 5 indeed failed on this chip, and is the failing vector mentioned in Figure 6(b). The shortlisting technique saves significant amount of run-time by avoiding further analysis.

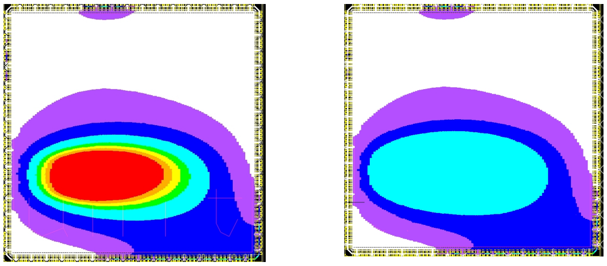

Figure 5: AstroRail Plots of IR-Drops for Chip-A: (a) for the failing vector (b) for the functional vector.

Figure 6 shows the AstroRail plots of the failing test vector for Chip-B. In Figure 6(a), the IR-drops are averages, calculated in a pattern-independent fashion. We see that the red area of Figure 6(a) shows potential problems due to IR-drop failure. In Figure 6(b), we plot the average IR drops again, but the averaging is done only for the vectors short-listed by our technique. As can be seen, the area marked red in Figure 6(b) is not only smaller, but has no overlap with the red area in Figure 6(a). This means two things: (1) A designer who looks at plot 6(a) and fixes the power rail to avoid excessive IR-drops may not really address the actual problem while still consuming costly silicon real-estate (2) In the example of Chip 6(b), the critical path that failed due to excessive IR drop indeed passes through the red area; recalling that this area has no overlap with the red area of Figure 6(a), we conclude that a vector-independent approach to IR-drop failure analysis can be misleading.

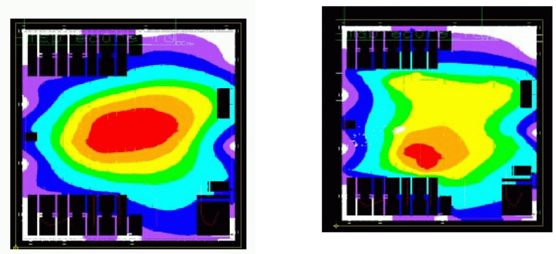

Figure 6: Astro-Rail Plots for Chip-B (a) Using Pattern-independent averages (b) Considering Pattern-dependent Average IR-drops

The typical run-times for TestRail are shown in Table 1 for different phases of the total flow. As can be seen, for a large design like Chip-B, the vector shortlisting can proceed at the rate of about 1 vectors/4sec. The toggle count estimation was done for the one shortlisted vector in case of Chip B, and this step took ~ 5 min. This clearly brings out the benefit of the vector shortlisting proposed in this paper; had we performed toggle count estimation for all the 80 test vectors, the total run time for even the toggle count estimation phase would have exceeded 3 hrs, not to speak of the run-times during Astro-Rail and timing analysis phases of the flow.

Table 1: Run-times for different phases of TestRail

	Vectors shortlisted	Vector Short-listing	Toggle Count Propn.	Power Rail Analysis	Critical Path Analysis
Chip-A	5	< 1 min	~ 5 min	~ 5 min	~ 20 min
Chip-B	80	~ 5 min	~ 5 min	~ 2 hrs	~ 1.5 hrs

6 Conclusions

In this paper, we discussed the problem of validating test vectors in the presence of IR drop problems. Although we discussed the validation of transition delay at-speed vectors, the technique is also valid for memory BIST and functional vectors, which are also run at-speed. We are aware of at least one design in our organization for which transition delay tests failed on the tester due the IR drop problems. Since it is late in the design flow to correct physical design of the power grid to fix the problem, in such an eventuality it would be necessary to discard these test vectors and generate alternate tests. We anticipate that a tool such as TestRail will be useful in such a situation. Since our approach is a static one, it is fast. Our method has three phases, namely, vector shortlisting, power rail analysis, and critical path analysis. The number of vectors decreases from one phase to next, resulting in reduced verification cycle time. In comparison to a simulation approach that required several days of effort to spot Vector 5 as a failing vector for Chip-B, our technique was able to spot it in a few hours. We are not able to compare our approach to any method in the literature since we are not aware of any other approach that directly competes with our work.

A layout-aware ATPG tool to generate correct transition delay tests that do not fail due to IR drop problems can also use the techniques reported in this paper. Such an ATPG tool will read in the power rail information and the threshold on the amount of toggling a power rail can tolerate. The ATPG can use several techniques to generate alternate test vectors: when ordering the test vectors and selecting the final vectors, power rail switching can be factored in. Alternately, switching activity can be reduced during test compression. In current industrial practice, ATPG tools are not able to take physical design information as input. Future work should focus on correct-by generation ATPG techniques which can avoid surprises down the test flow. In the absence of ATPG that is aware of DSM effects, there will always be a need to verify test vectors. If this verification cannot comprehend all the DSM effects, there is a tendency to overdesign to avoid problems. However, due to the impact of overdesign on chip yield, designers are wary of using large design margins. The technique presented in this paper is a design aid that can help save precious silicon area without significantly impacting verification time.

Acknowledgments

We wish to thank Anil Kumar Gajkela, M. Thenappan, Charu Srimali and Supatra Basu of Texas Instruments India, and Murugesh of Tata Elxsi, for their help in running the experiments. We also thank Arvind, Harinath, PS Vijay Kumar, Ananth Somayaji, J. Sridhar, and Suravi Bhowmick for their helpful comments in the early stages of this work.

References

[1] G. Bai, S. Bobba, and I.N. Hajj. *"Power Bus Maximum Voltage Drop in Digital VLSI Circuits."* In Proceedings of International Symposium on Quality of Electronic Design, pages 263-268, March 2000.

[2] D. Bhattacharya, P. Agrawal, V.D. Agrawal, "*Test generation for path delay faults using Binary Decision Diagrams*," IEEE Transactions on Computers, March 1995

[3] Michael L. Bushnell, Vishwani D. Agrawal. Essentials of Electronic Testing for Digital, Memory, and Mixed-Signal VLSI Circuits, November 2000

[4] R. Chaudhry, D. Blaauw, R. Panda, T. Edwards, "*Current signature compression for IR-Drop analysis*", Proceedings of IEEE design Automation conference, 2000.

[5] D.I. Cheng, K.T. Cheng, D.C. Wang, M.M. Sadowska. "Hybrid methodologies for switching activities estimation." IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. Vol. 17, No.4, April 1998.

[6] R. Dutta and M. Marek Sadowska. *Automatic Sizing of Power and Ground Networks in VLSI*. Proceedings of the 26th IEEE Design Automation Conference, June 25-29, 1989.

[7] Haroun, A.J., F. Routabi, B. Khalili, "*Power estimation tool for sub-micron VLSI circuits.*". In IEEE International Conference on Computer Aided design, 1992

[8] H. Kriplani, F. Najm, and I. Hajj. *Resolving signal correlations for estimating maximum currents in CMOS combinational circuits*. In Proceedings of 30th IEEE/ACM Design Automation Conference, pages 384-388, 1993.

[9] H. Kriplani, F. Najm, and I. Hajj. *Worst Case Voltage Drop in Power and Ground Busses in CMOS VLSI Circuits. SRC TECHON'93 Conference*, Atlanta, GA, pp. 402-404, September 28-30, 1993.

[10] A. Krstic, K. T. Cheng "Vector generation for maximum instantaneous current through supply lines for CMOS circuits", Proceedings of IEEE design Automation conference, 1997

[11] L R Liu, H.C. Chen, H. C. Du, "*The calculation of signal stable ranges in combinational circuits*", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, August 1994

[12] Synopsys, Inc. *Prime Time User Manual*, 2002.

[13] Synopsys, Inc. *Power Compiler User Manual*, 2002.

[14] Synopsys, Inc. *AstroRail User manual*, 2002.

[15] Texas Instruments India. Toggle Frequency Calculator Reference Manual, 2003.