PATTERN-INDEPENDENT CURRENT ESTIMATION FOR RELIABILITY ANALYSIS OF CMOS CIRCUITS

by

Richard Burch, Farid Najm
Ping Yang, and Dale Hocevar

ABSTRACT

Accurate and efficient expected current estimation is required in circuit designs to analyze electromigration failure rate, power consumption, voltage drop, etc. A new pattern-independent simulation approach for estimating this expected current waveform drawn by CMOS circuitry has been developed. Four original concepts, *probability waveforms*, *probability waveform propagation*, *probabilistic circuit models*, and *statistical timing analysis*, are presented which allows an efficient and accurate estimation of expected current waveforms. This approach is dramatically faster than traditional methods and yields comparable results.

Topics : 1, 16, 12.

I. INTRODUCTION

The quality of an integrated circuit is measured by both functional and reliability standards. Many simulation approaches exist to verify that a design will meet functional specifications; however, present capabilities for verifying that a design will meet reliability specifications are extremely limited. At Texas Instruments, much effort is being devoted to providing the simulation tools and expertise to verify, prior to manufacturing, that chips will satisfy reliability specifications. This is part of an overall strategy to reduce design cycle time.

A major reliability problem, electromigration, has been the subject of substantial modeling effort [1]. These models utilize the current density in a metal section to predict its median time to failure due to electromigration. A simulation tool, SPIDER [2], has been developed that will estimate the median time to failure for each section of metal corresponding to any interconnect signal that the user designates. It requires the user to specify current sources to load the signal at specified connection points. Using these contact points as electrical nodes, SPIDER extracts an equivalent resistance network to represent the metal sections of the interconnect signal, simulates the network using SPICE [3] to determine the current density in each section, and then estimates the median time to failure of each section.

This approach works well; however, the user must obtain current sources to model the loading of the extracted network. In CMOS, one serious electromigration problem occurs on power (V_{dd}) and ground (V_{ss}) lines; a method of automatically providing the current sources to load these lines is necessary. Another problem associated with current density on power (or ground) lines is excessive voltage drop (or rise) in the voltage on the line, caused by the resistance of the power (or ground) line. Excessive voltage drop can prevent CMOS chips from meeting functional specifications; however, since it is a chip level phenomenon, traditional simulation tools do not detect the problem. SPIDER analyzes voltage drop as well as predicting electromigration failures, but current sources are still needed to model the loading of the circuitry. To provide these sources, either all possible input vectors need to be exercised to derive a statistically meaningful "expected" current waveform or a particular input vector need to be used to obtain the worst case current waveform. In general, the all possible input vector approach with SPICE is computationally prohibitive and the designers cannot provide the worst case input vector. This paper presents a new simulation approach developed to estimate the necessary expected current waveforms, for V_{dd} and V_{ss}, of CMOS circuits. This approach is much faster than SPICE, especially when the user is uncertain about the correct circuit inputs.

II. BASIC CONCEPTS

Current sources are required to model circuit operation under normal conditions (for electromigration) or worst case conditions (for voltage drop). It is very difficult for the user to specify input waveforms for either condition without examining a variety of inputs. For a circuit with n inputs, the number of possible input patterns is 2^n and the number of possible transitions at the inputs is 2^{2n}. An analysis of the current drawn for a given input is feasible, although fairly expensive for a large circuit, using a circuit analysis program such as SPICE. It is clear that redoing this analysis for all possible inputs is prohibitively expensive. Any approach that avoids this *pattern-dependence* problem is called *pattern-independent*.

Even if one derives the *worst case* current drawn by small subcircuits inside a big circuit, it is difficult to combine these currents to get the overall worst case current. A simple sum is grossly inaccurate because the input that causes the worst case total current may cause internal subcircuits to draw less than their maximum. An exact search algorithm that searches for this particular input is also prohibitively expensive.

While a few published articles have looked at the average power consumption problem, very little work has been done on the *current estimation* problem. The authors know of one recent work by Tyagi [4] that tackles this problem for CMOS circuits. The pattern-dependence problem is avoided in [4] by borrowing the concept of a *stage* from timing verifiers such as TV [5] and CRYSTAL [6]. A stage is basically a series path of transistors connecting a power or ground node to an output node. Based on this concept, Tyagi derives the charging current drawn by a single gate. This approach is simple enough to be efficient for large circuits, but is not sufficiently accurate. The stage is important in timing verification because it causes the worst timing delay and, therefore, the *least* current to be drawn, not the worst current.

We present a new technique for deriving a pattern-independent estimate of the power supply and ground currents drawn by a CMOS circuit. To achieve pattern-independence, accurate information about the current is sacrificed and only *statistical* information is derived. More precisely, if one envisions the set of all possible transitions at the inputs of a CMOS circuit, one can capture information about this set by considering each transition as an event with a certain probability and then extracting statistical information about the current drawn based on these probabilities. Over this probability space, the supply (or ground) current waveform becomes a *stochastic process* [7]. An important *second-order property* used to describe a stochastic process is its *mean*. In this case, the mean is a current waveform whose value at each time point is the expected value or mean of all the values that the actual current can take at that time. It is precisely this *mean* that the approach we present here can derive, and we will call it the *expected current waveform*. Future extensions of the technique will aim at deriving another second-order property of the current, namely its variance or standard deviation as a function of time.

We emphasize that the expected current waveform is not the same as a time-average of the current, it is a waveform which, if the standard deviation of the actual current values is small, will approximately match the real current waveform(s). Under this condition, it can be used to estimate electromigration failure rates and voltage drops. If the standard deviation is large, this waveform remains satisfactory for estimating electromigration median time to failure (MTF). Using the electromigration model in [1], it can be shown that MTF is related either to the expected waveform of J (the current density) or that of J^2 (which we estimate, having found the expected waveform of J). This is the motivation and justification of our approach.

To derive the expected current waveform, $E[i(t)]$, we build on the concept of *signal probabilities* [8], which has recently become popular in the testing field [9]. Given that different input patterns may occur and that every pattern has a certain (user defined) probability of occurrence, then every input signal line acquires a certain probability of being high (or low). Internal circuit node probabilities can be derived from the input probabilities given the connectivity of the circuit. This concept has recently been used to estimate the power consumption of CMOS circuits [10]. The power consumption is closely related to the time-average of the current which, as pointed out above, is different from our $E[i(t)]$. If P_{ih} is the probability that node i is high, then the probability of a transition at i is taken to be $P_{ih} \times (1 - P_{ih})$ in [10]. This assumes that the signal probability at i before a transition is the same as after, and that the two signal values are independent random variables. In this work we make no such assumptions, we extend the signal probabilities concept to include *transition probabilities*. The transition probability of a signal i at time t is the probability of a low-to-high transition from t^- (just before t) to t^+ (just after t), denoted $P_{ilh}(t)$. Given these probabilities at the inputs then internal node transition probabilities can be derived in the same way as signal probabilities.

Using transition probabilities we can compactly describe a large set of logic waveforms. For example : input i is high with probability .76 at time 0 ($P_{ih}(0) = .76$), switches low-to-high with probability .5 at time 2ns ($P_{ilh}(2ns) = .5$), is then high with probability .35 at time 3ns ($P_{ih}(3ns) = .35$), etc... . The circuit inputs are described by giving such waveforms for every input node, the resulting input data is a sequence of vectors of probability values, which we will refer to as *probability vectors*. It's clear then that the signal probability concept is now extended to a *probability waveform* concept. Our probabilities are themselves waveforms, functions of time. Formally speaking, such a probability waveform defines a discrete stochastic process, whose individual outcomes are the different possible logic waveforms. These powerful new concepts provide the originality in our approach and make possible the derivation of accurate expected current waveforms.

III. PROBABILITY PROPAGATION AND CURRENT ESTIMATION

Our approach can be summarized as follows :

-1- Given probability waveforms at the primary circuit input nodes, derive the corresponding waveforms at internal circuit nodes, this is *probability propagation*.

-2- With some convenient partitioning of the circuit examine every subcircuit and derive its expected current waveform $E[i(t)]$ based on the waveforms at its inputs.

-3- Add these current waveforms to get the total expected current waveform for the circuit. This step is valid because $E[i_1(t_0) + i_2(t_0)] = E[i_1(t_0)] + E[i_2(t_0)]$, $i_1(t_0)$ and $i_2(t_0)$ are random variables.

Steps 1 and 2 constitute the major tasks in our technique, we will try in what follows to explain the work and the problems involved in them.

III-A. Probability Propagation

For static signal probabilities (as opposed to probability waveforms) propagation has been addressed by several researchers. The solution in [8] uses symbolic analysis, that in [9] gives an algorithm with several suggested heuristics to improve efficiency, while [11] suggests an efficient fresh approach that gives bounds on the probabilities rather than exact probabilities. The problem is of linear complexity for circuits with no reconvergent fanout or feedback, but becomes exponential otherwise. In fact, it's easy to prove that if reconvergent fanout and feedback are allowed then the problem becomes $\mathcal{NP}-hard$. While the inputs to the circuit can be assumed to be independent random variables, the existence of reconvergent fanout or feedback causes internal nodes to be dependent, which complicates the propagation problem. We have decided to adopt the approach in [9] as a first implementation. According to this approach, a *supergate* is defined as a *minimal* subset of the circuit with independent inputs. If the supergate is a single gate, then the gate's inputs are independent, and propagation through it is simple as detailed below. If the supergate has more than one gate, then some of the inputs to the gates are dependent. Propagation through such a supergate is done by considering all combinations of vectors at some (or all) of its inputs. For each of these combinations, the internal gates of the supergate have independent inputs and propagation through them is simplified. A variety of heuristic will be examined to reduce the complexity in such cases.

For probability waveform propagation, the problem is even more complicated because of the extra time dimension involved. The time dimension actually creates another kind of dependency problem; for example, if $P_{ilh}(t) = 0$ (or 1) then the signals $i(t^-)$ and $i(t^+)$ are dependent random variables, equal (or opposite) in this case. Such dependency can also arise in other situations, and needs to be properly dealt with. This kind of dependency will be called *temporal dependency* while that described above, caused by reconvergent fanout or feedback, will be called *spatial dependency*.

The propagation issues discussed so far have been related to the global problem of propagation through a gate-level description of the circuit. We now discuss propagation through a single gate. The approach in [10] involves assigning directions to the transistors and then propagating the node probabilities along these directions, starting at the power or ground nodes (which have known probabilities), to the gate output node. Even if the gate inputs are really (spatially) independent this approach neglects dependency resulting from reconvergent fanout of the assigned directions inside the gate, such as a transistor in series with a parallel combination of two other transistors.

We propose the following new solution to this problem. Using the supergate approach [9], we can limit our attention to gates with spatially independent inputs as explained above. Given such a CMOS gate, we want the probability that its output is high (or transitions), knowing the signal (or transition) probabilities at its inputs. Represent the p (or n) block of the gate as a graph. Each MOSFET generates a graph edge with a probability of being on (or transitioning off to on) determined by the signal (or transition) probability at its gate. We reduce the graph to a single equivalent edge between the output and V_{dd} (or V_{ss}). This edge contains the probability that there is a conducting path between the output and V_{dd} (or V_{ss}), and the probability that this path goes from off to on. If the graph is series/parallel, then it's a simple matter to perform series/parallel reductions that generate the equivalent edge, the primitive operations involved are those of combining two transistors in series (intersection of two events) and combining two transistors in parallel (union of two events), and can be carried out using simple probability theory. As discussed above, the approach in [10] suffers from a dependency problem in series/parallel networks. Our approach has no such problem. For bridge circuits we make use of the *node elimination* technique which is the graph-domain operation corresponding to Gaussian Elimination on the graph's adjacency matrix [12]. These eliminations can introduce the same dependency problem because an edge may be split into two or more edges. Eventually, solutions to this problem will be investigated.

This propagation is done every time one or more of the signal probabilities at the inputs of the gate changes, and constitutes probability waveform propagation.

III-B. Current Estimation

With the ability to propagate signal and transition probabilities through each gate of the circuit, we are now ready to discuss the current estimation procedure at each gate. We will focus on standard CMOS fully complementary gates; this is the extent of our first implementation and is being extended to more general circuits. Furthermore, a gate will be assumed to have spatially independent inputs. The general case is properly handled using the concept of a supergate, described above, with the independent-inputs-gate-solver used as a subroutine. If at least one input has a non-zero transition probability at t, the gate draws a current pulse depending on the specific input node and transition; we'd like to find the expected waveform or pulse of all these possible pulses. The expected gate current pulse will be modeled by a triangular pulse that starts with a peak of $E[pk\ i] \doteq E[i(t^+)]$ at time t and decays linearly to zero at time $t+\tau$. In what follows we'll explain how $E[pk\ i]$ and τ are derived.

The analysis given here will focus on the charging current component and leave out the *direct* component which may be drawn through a path of p and n channel transistors during the transition. We have adopted this policy as a first implementation based on Veendrick's [13] work which basically says that if the gate is "well designed" then the direct current component may be neglected. In the future, we will extend our approach to include this current.

Consider the generic CMOS gate structure shown in Fig. 1. The figure shows the p-transistor block, the n-transistor block, and the output node capacitance split into two lumped capacitors C_p to V_{dd} and C_n to V_{ss}. Similarly, each internal node n_i has two capacitances C_{in} and C_{ip}. On a low-to-high transition, the currents flowing through C_n and C_p at the output node are i_{p1} and i_{p2}, respectively, as shown in the figure. The corresponding i_{n1} and i_{n2} for a high-to-low transition are also shown. The currents i_{p2} and i_{n2} are discharging currents that redistribute locally, and we are interested in $i = i_{p1} + i_{n1}$. Of course these currents are only to the output node and the total gate current i_{tot} will be larger than i, however the output current plays a central role in the derivation.

Let $i_p = i_{p1} + i_{p2}$ and $i_n = i_{n1} + i_{n2}$. It's easy to verify that $i_{p1} = i_p \times (C_n \ / \ C_p + C_n)$, and $i_{n1} = i_n \times (C_p \ / \ C_p + C_n)$. Therefore :

$$E[i(t)] = E[i_p(t)] \times \frac{C_n}{C_p + C_n} + E[i_n(t)] \times \frac{C_p}{C_p + C_n} \tag{1}$$

And in particular, the value at the peak is :

$$E[pk\ i] = E[pk\ i_p] \times \frac{C_n}{C_p + C_n} + E[pk\ i_n] \times \frac{C_p}{C_p + C_n} \tag{2}$$

The values of $E[pk\ i_p]$ and $E[pk\ i_n]$ are derived as follows. For i_p, consider the p part of the gate, and let every transistor T_k be represented by a switch of on-conductance g_k, where $V_{dd} \times g_k$ is the current that flows in T_k with V_{dd} volts across it (drain to source) and 0 volts at its gate (T_k in saturation). This is taken to be the definition of g_k because we're interested in the peak current drawn, which occurs during saturation. We derive g_k from the transistor model parameters given by the user. Now let $G_p(t)$ be the conductance between the output node and V_{dd}, $G_p(t)$ is a random variable (at every time point t) and depends on which transistors are actually on. If an event occurs at the gate at time t, then the value of $E[G_p(t^+)]$ and the previous state of the output node, $V_o(t^-)$, will determine $E[pk\ i_p]$. Formally, we have $E[pk\ i_p] = E[(V_{dd} - V_o(t^-)) \times G_p(t^+)]$, which becomes :

$$E[pk\ i_p] = V_{dd} \times E[G_p(t^+)|G_p(t^-) = 0] \times P(G_p(t^-) = 0) \tag{3}$$

Where $E[A|B]$ is the *conditional expected value* of A given B, and $P(B)$ is the probability of the event B. The formula is correct because $G_p(t^-) = 0$ (1) means $V_o(t^-) = 0$ (V_{dd}). Similarly for the n part of the gate, we get :

$$E[pk\ i_n] = V_{dd} \times E[G_n(t^+)|G_n(t^-) = 0] \times P(G_n(t^-) = 0) \tag{4}$$

The value of $E[G_p]$ (or $E[G_n]$) may be derived from the graph as follows. If the gate terminal of every transistor T_k is x_k, and if G_k is the random variable describing the transistor conductance, then $E[G_k] = g_k \times P_{x_k l}$ for a p transistor, and $g_k \times P_{x_k h}$ for an n transistor. Consider the graph for the p part of the gate with each edge labeled with its $E[G_k]$. Perform series/parallel reductions and node eliminations to reduce the graph to a single edge so that at every step in the reduction the value $E[G]$ between the output node and V_{dd} remains unchanged. The same can be done for the n part of the gate.

Now, to find the *conditional* expected value of G_p, $E[G_p(t^+)|G_p(t^-) = 0]$, we perform the same graph reduction using $E[G_k(t^+)|G_p(t^-) = 0]$, instead of $E[G_k(t^+)]$, for every transistor. If the values of the inputs at time t^+ are independent of their values at t^- then $E[G_p(t^+)|G_p(t^-) = 0] = E[G_p(t^+)]$ and the problem would be simplified. However this is not true in general and the values $E[G_k(t^+)|G_p(t^-) = 0]$ need to be computed every time the gate gets an event. Exactly how these are computed will not be included here for lack of space, suffice it to say that it takes one graph reduction to evaluate each of them.

Having found $E[pk\ i]$ for the output node, the expected value of charge delivered to (or from) the output node capacitors is easily found as follows :

$$E[q] = V_{dd} \times C_n \times P_{olh}(t) + V_{dd} \times C_p \times P_{ohl}(t) \tag{5}$$

where o is the output node. We now make the assumption that the time constant for charging the output node is the largest of the internal gate nodes. We let the time span of the output node current represent the time span τ of the total gate current. By the triangular pulse approximation :

$$\tau = 2 \times \frac{E[q]}{E[pk\ i]} \tag{6}$$

Next the expected value of the charge delivered by the total gate charging current, $E[q_{tot}]$ is derived using the capacitances at each internal node j as follows :

$$E[q_{tot}] \approx \sum_{j \in p\ block} V_{dd} C_{jn} P_{jlh} + \sum_{j \in n\ block} V_{dd} C_{jp} P_{jhl} \tag{7}$$

Strictly speaking the probabilities P_{jlh} and P_{jhl} are hard to find. We have therefore opted to use an upper bound of these probabilities to replace them in the equation. An upper bound of P_{jlh} (P_{jhl}), for a node in the p (n) block, is the probability that the conduction state between j and V_{dd} (V_{ss}) goes from off-to-on. This is found by a graph reduction that repeats the work done to find P_{olh} for the output node o for every internal node j. Finally, the peak total current is found as :

$$E[pk\ i_{tot}] = \frac{2E[q_{tot}]}{\tau} = \frac{E[q_{tot}]}{E[q]} \times E[pk\ i] \tag{8}$$

Having derived the expected gate current pulse, the new event at the output of this gate (the possible occurrence of a transition) needs to be properly placed in time. The time of this event can be derived follows. If one considers a resistor R charging a capacitor C from a power supply V, then the current and voltage at the capacitor both reach their half-point at time $.693 \times RC$. If we assume that the individual current pulses $i(t)$ are exponentially decaying, rather than linear, then the switching time at the output, t_s, is $0.693 \times (C_p + C_n)\ /\ G_p$ for a low-to-high transition, and $0.693 \times (C_p + C_n)\ /\ G_n$ for a high-to-low transition. This t_s is, again, a random variable, and one is interested in $E[t_s | V_o$ transitions$]$. Knowing that the duration of i_p (or i_n) determines the gate delay, and that these currents deliver charge to *both* C_p and C_n, then if q_p and q_n are the charges delivered, we have :

$$E[q_p] = V_{dd} \times (C_p + C_n) \times P_{olh} \text{ , and} \tag{9}$$

$$E[q_n] = V_{dd} \times (C_p + C_n) \times P_{ohl} \tag{10}$$

The duration of the two pulses is derived as before, as :

$$\tau_p = 2 \times \frac{E[q_p]}{E[pk\ i_p]} \text{ , and } \tau_n = 2 \times \frac{E[q_n]}{E[pk\ i_n]} \tag{11}$$

Having found these values, the time delay is :

$$E[t_s | V_o \text{ transitions}] = 0.35 \times \frac{\tau_p \times P_{olh} + \tau_n \times P_{ohl}}{P_{olh} + P_{ohl}} \tag{12}$$

It is important to note that τ_p and τ_n are independent of the particular partitioning of $(C_p + C_n)$, which makes the timing estimate reliable.

IV. PROBABILISTIC SIMULATION FOR CURRENT ESTIMATION

Now that we have developed the capabilities to propagate signal and transition probabilities, and to estimate a gate's expected current pulse and its delay for these probabilities, we are ready to integrate these tools into a useful simulator for estimating the expected current waveform drawn by CMOS circuitry under normal operating conditions. In this section, we describe the input requirements of such a simulator, the partitioning algorithm, the overall simulation approach, and the modifications to the basic simulator required to accurately handle spatial dependency and feedback.

The proposed simulator would require three forms of input from the user. First, the user must supply a SPICE, or equivalent, description of the CMOS circuitry for which he wishes to estimate the current drawn. Second, he must supply a list of the nodes which represent connections to the power lines of the chip and a list of the nodes that represent ground connections. Finally, he must supply probability vectors (see section II) for nodes that are primary inputs to the circuitry.

The first step is to construct a representation of the circuit from the SPICE input, with any subcircuit hierarchy removed. Two structures are used to represent the circuit: elements and nodes. For each MOSFET in the SPICE deck, an element is created. The element joins the source and drain nodes of the MOSFET; these become the connection nodes of the element. It is composed of a switch in series with a conductance. The switch is controlled by the gate node of the MOSFET; this node is the input node of the element. Nodes are used to represent the electrical nodes of the circuit. In addition to containing connectivity information, each node contains an estimate of the total capacitance connected to the node, including MOSFET capacitances. This capacitance is used in current estimation.

Next, a simple partitioner joins together all elements that have common connection nodes, that are not power or ground nodes, to form gates. The circuit is now represented by a network of gates joined together by nodes.

Once the partitioned circuit has been constructed, it must be simulated over a time interval specified by the user to estimate the expected current waveform during that interval. Current estimation by the methods we propose is similar to logic and timing simulation, and we borrow the event driven simulation approach from there. Each time there is a non-zero chance of a transition on a node, an event is created. These events are kept in an event queue, and store the transition time, the probability of transition, and a pointer to the affected node. When an event occurs, all gates that have the event's node as an input are examined. The expected current pulse, the gate delay, and the signal and transition probabilities at the output node of each gate are calculated with methods previously described. A new event is added to the event queue at the current time plus the gate delay, indicating a transition on the output node with the signal and transition probabilities determined during propagation. The process is repeated until the event queue is empty of events that fall within the specified simulation interval.

Since the gate takes a finite amount of time to draw its current, events that are very close together on different inputs of the gate should be treated as simultaneous. When a gate is simulated for a scheduled event, the next events that affect the other inputs of the gate are checked. If any of them occur before the output would transition, estimated from a reference gate delay, then that event is considered to occur simultaneously with the scheduled event for this gate only.

Spatial dependency presents particular problems, as previously discussed. Spatially dependent sections of the circuit must be split into supergates, whose inputs are independent. These supergates must then be simulated for every possible combination of transitions on the nodes that caused the spatial dependency inside the circuit. This can be simplified since events that no longer have any effect on the supergate do not require separate simulations; however, the complexity is still exponential. Various heuristics are being investigated to determine if an acceptable compromise between accuracy and simulation simplicity can be found for supergates.

V. RESULTS

The probabilistic simulation approach just described has been implemented in a program called CREST (CuRrent ESTimation program). Circuits with spatial dependency are not allowed, but this capability is presently being added. Our goal is to approximate the expected pulse drawn by the circuit for known signal and transition probabilities on the inputs of the circuit. For comparison, we generated the expected current pulse for a variety of examples by running SPICE on every input voltage waveform allowed by the probability vectors, weighting each pulse by the probability that the waveform producing it would occur, and summing the weighted pulses to produce the expected pulse. Figs. 2–6 show some of the results obtained. When the circuit title is not sufficient to define the circuit, a sketch is included.

In all examples tested, the results were excellent. Peak currents were within 20%, average currents were within 10%, and, as clearly shown in Fig. 6, timing estimates were within 10% of SPICE.

To be used to estimate current of today's large VLSI designs, this approach must be much faster than SPICE. Table 1 compares CREST simulation speed to SPICE simulation speed for all vectors necessary to generate the expected pulse and the SPICE simulation speed divided by the number of vectors. In overall simulation speed CREST is dramatically faster than SPICE, due to the exponential number of vectors that must be simulated for multiple inputs. CREST simulation speed is still significantly faster than the SPICE time per vector and the trend we observed was for the speed gain of CREST over SPICE to grow with circuit complexity (Fig. 7). Since each test circuit had to be specially designed to prevent spatial dependency, only relatively small examples have been tested.

VI. CONCLUSION

In this paper, we have described a pattern-independent simulation approach for current estimation. This approach provides expected current waveforms appropriate for predicting the median time to failure from electromigration, analyzing voltage drop, and estimating power consumption. The speed for simulation of a single input vector is dramatically faster than SPICE. Moreover, the combined effects of a variety of input patterns, each of which would require separate SPICE simulations, can be derived with a single simulation at no more expense than the simulation for a single vector. As such, our approach is pattern independent and provides a dramatic speed-up over the the conventional approach using SPICE. The waveforms produced with our approach agree well with results obtained from SPICE: peak currents are within 20%, average currents are within 10%, and timing estimates are within 10%.

CREST will be expanded to include the algorithms necessary to handle spatial dependency and more complex gates, such as pass transistor structures. Evaluation of the complete approach can then be performed, and it can be extended to other technologies; NMOS, bipolar, etc.

VII. ACKNOWLEDGEMENTS

The authors would like to thank Joe Hall, Mike McGraw, and the Design Automation Department at Texas Instruments for their invaluable support and contribution to this work.

REFERENCES

[1] J. W. McPherson, P. B. Ghate, and P. Lou, "Design implications of transient induced electromigration failures," *Report # 03-83-30, Semiconductor Group, Texas Instruments Inc.*, October 1983.

[2] J. E. Hall, D. E. Hocevar, P. Yang, and M. J. McGraw, "SPIDER – a CAD system for modeling VLSI metallization patterns," *IEEE Transactions on Computer Aided Design*, vol. CAD-6, pp. 1023–1031, Nov. 1987.

[3] L. W. Nagel, "SPICE2: a computer program to simulate semiconductor circuits," *PhD thesis, Dept. of Elec. Eng., Univ. of California, Berkeley*, 1975.

[4] A. Tyagi, "Hercules: a power analyzer for MOS VLSI circuits," *IEEE International Conference on Computer Aided Design*, November 1987.

[5] N. P. Jouppi, "Timing analysis for nMOS VLSI," *Proceedings of the* 20^{th} *Design Automation Conference*, Miami Beach, pp. 411-418, June 1983.

[6] J. K. Ousterhout, "A switch-level timing verifier for digital MOS VLSI," *IEEE Transactions on Computer Aided Design*, vol. CAD-4, no. 3, pp.336–349, July 1985.

[7] A. Papoulis, *Probability, Random Variables, and Stochastic Processes.* New York, NY: McGraw-Hill Book Co., 1984.

[8] K. P. Parker and E. J. McCluskey, "Probabilistic treatment of general combinational networks," *IEEE Transactions on Computers*, pp. 668–670, June 1975.

[9] S. C. Seth, L. Pan, and V. D. Agrawal, "PREDICT – probabilistic estimation of digital circuit testability," *IEEE* 15^{th} *Annual International Symposium on Fault–Tolerant Computing*, Ann Arbor, MI, pp. 220–225, June 1985.

[10] M. A. Cirit, "Estimating dynamic power consumption of CMOS circuits," *IEEE International Conference on Computer Aided Design*, November 1987.

[11] J. Savir, G. S. Ditlow, and P. H. Bardell, "Random pattern testability," *IEEE Transactions on Computers*, vol. C-33, no. 1, pp. 79–90, January 1984.

[12] S. Parter, "The use of linear graphs in gauss elimination," *SIAM Review*, vol. 3, no. 2, pp. 119–130, April 1961.

[13] H. J. M. Veendrick, "Short-circuit dissipation of static CMOS circuitry and its impact on the design of buffer circuits," *IEEE Journal of Solid-State Circuits*, vol. SC-19, no. 4, August 1984.