# CREST - A CURRENT ESTIMATOR FOR CMOS CIRCUITS

by

Farid Najm†, Richard Burch‡, Ping Yang‡, and Ibrahim Hajj†

†Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

‡VLSI Design Laboratory
Texas Instruments Inc.
Dallas, Texas 75265

## ABSTRACT

CREST is a pattern-independent current estimation approach developed to support electromigration analysis tools. It uses the powerful, original concept of *probabilistic simulation* to efficiently generate accurate estimates of the expected current waveforms. As originally implemented, CREST was restricted to circuits with no pass transistors and no reconvergent fanout or feedback. This paper extends the approach to circuits containing pass transistors, reconvergent fanout, and feedback, and provides heuristics to efficiently simulate circuits with large reconvergent fanout or feedback blocks. The results of CREST on several real circuits are presented.

## I. INTRODUCTION

The reliability of integrated circuits is quickly becoming a major concern for the electronics industry; this creates a need for efficient and accurate CAD tools that guarantee design reliability. This work is concerned with one important aspect of the reliability problem, electromigration (EM). EM failures are caused by the transport of metal atoms due to electron flow. The failure rate depends on the current density in the metal lines and is expressed as median-time-to-failure (MTF). The first step in any scheme to estimate the MTF of the metal lines of a chip requires an estimate of the current drawn by the circuitry. Since EM is a function of the combined effects of all current density waveforms encountered during operation [1], a current waveform must be found which accurately models these effects.

In [2] we described a new technique to solve the current estimation problem. The method was restricted to circuits with no pass transistors and no reconvergent fanout. This paper extends the techniques described in [2] to remove these limitations. The techniques have been implemented in the program CREST, which has excellent performance in both accuracy and speed. We present two heuristics that boost the performance in the case of reconvergent fanout, and thus make it more acceptable for VLSI circuits.

## II. BACKGROUND

The currents in CMOS are directly related to the specific circuit inputs. This raises an important question. What is the input sequence that gives a representative current waveform for EM analysis? One approach would be to use all possible current waveforms when estimating MTF. This would be prohibitively expensive, since a separate simulation would be required for each possible input pattern.

To avoid this computationally expensive problem, we tackle a related problem : given a user-specified range of possible inputs, derive statistical information about the corresponding current waveforms. One desirable feature of this approach is that it is *pattern-independent* [2], requiring minimal input from circuit designers.

Replacing logic values by *signal probabilities* [3] and logic transitions by *transition probabilities* [2], CREST performs an event-driven *probabilistic simulation* of the circuit. An event at a node $x$ is specified by four parameters : the time of transition $t$, and the probabilities that $x$ is high before $t$ ($P_h^-$), $x$ transitions from low to high at $t$ ($P_{l-h}$), and $x$ is high after $t$ ($P_h$). Given such user-specified events at the primary inputs, CREST propagates the corresponding *probability waveforms* into the circuit using its transistor-level description. It also derives, for every gate, the expected value at every time point of the supply and ground currents. These values are summed to give an *expected current waveform* at every contact point on the bus. To propagate waveforms through a gate, a graph is created with each transistor forming an edge, containing the probability that its channel is on, and transitioning from off to on. This graph is reduced to a single edge between output and power that gives the probability that the output is high (edge is on), and the probability that the output is transitioning from low to high (edge is transitioning from off to on). The current pulse for a gate is determined by performing similar graph reductions.

## III. PASS TRANSISTOR SIMULATION

Pass transistors present a problem in the probabilistic simulation approach used in CREST [2]. Current drawn through pass transistors is a relatively small fraction of the overall current dissipated in a CMOS circuit and can probably be ignored. However, transition events must be propagated correctly through them. One previous assumption in CREST was that the output of a gate always has a conducting path to either power ($V_{dd}$) or ground ($V_{ss}$). If a gate contains pass transistors, this assumption is not valid and can produce very inaccurate estimates (Fig. 2). We present a new scheme for propagating probability waveforms through pass transistor gates that does not make this assumption.

Two types of gates are needed for analysis of pass transistor sections: *stages* and *wires* (Fig. 1). Stages represent conducting paths that connect two nodes in the pass transistor section; while, wires are used to tie together the outputs of two or more stages. These gates can be used to build complicated pass transistor networks. The direction of each gate must be carefully assigned, and we use several levels of algorithms, including some rules from [4].

Nodes in a pass transistor section of a circuit can have four valid states: conducting path to $V_{dd}$ (high tied or ht), conducting path to $V_{ss}$ (low tied or lt), charged with no conducting paths (high floating or hf), and discharged with no conducting paths (low floating or lf). An additional state is needed to eliminate dependency problems between the inputs of wires: no path to either $V_{dd}$ or $V_{ss}$ (floating or x). Although there is redundant information, five probabilities are used to fully define the states of all nodes in a pass transistor structure: $P_{ht}$, $P_{lt}$, $P_{hf}$, $P_{lf}$, and $P_x$.

Since $P_{hf}$ and $P_{lf}$ depend on previous states, they cannot be determined statically; however, if an initial state is assumed, the probabilities can be updated through *expanded transition events*. A normal transition event contains the probability that the input was high ($P_h$) before and after the transition and the probability of a transition from low to high ($P_{l-h}$). Expanded transition events must contain the probability that the node was high tied ($P_{ht}$), low tied ($P_{lt}$), and high ($P_h$) before and after the transition. Additionally, transition probabilities from low to high ($P_{l-h}$), low tied to high tied ($P_{lt-ht}$), low tied to floating ($P_{lt-x}$), floating to high tied ($P_{x-ht}$), and floating to floating ($P_{x-x}$) are needed to propagate probability waveforms through the pass transistors. Formulas have been derived for each probability for both stages and wires.

The propagation formulas for a stage depend on the probability waveforms at the source/drain input node and the probability that the stage is on, or transitioning from off to on. A stage is on if there exists a conducting path between its source/drain input and its output, and the probabilities are determined by a graph reduction similar to the one presented in the background section.

The propagation formulas for wires depend on the transition and signal probabilities at their inputs. Two minor assumptions must be made in the derivation of these formulas; however, this error is much less than assuming that the output is complementary (Fig. 2).

Accurate probability waveform propagation requires correct placement of transition events (Fig. 4). Although not the most accurate timing estimate, the approach used by Horowitz [5] is suited to our probabilistic models. By altering transition events so that they store the conductance used to produce the event and calculating the conductance from the stage's source/drain input to each node, the time constant can be accurately estimated.

## IV. SUPERGATE SIMULATION

The probability waveforms used in CREST are an extension of the concept of *signal probabilities* [3], which has recently become popular in the testing field [6]. Our current estimation algorithms [2] require that the probability waveforms, at the inputs of each gate, be *independent*. All primary circuit inputs are assumed to carry independent random variables. Thus, two internal circuit nodes are independent if they do not depend on a common primary input, otherwise they are *dependent*. We borrow the concept of *supergates* from [6]; a supergate is simply a *minimal* subset of the circuit with independent inputs. Non-trivial supergates are caused by feedback and reconvergent fanout. Any supergate input nodes that are reconvergent fanout stems, or that affect internal reconvergent fanout stems of the supergate, are called *reconvergent fanout input nodes*, abbreviated *rfi* nodes.

If logic (not probability) waveforms are assigned to rfi nodes, then all internal supergate nodes become independent and the supergate can be easily simulated. If the current waveforms are generated separately for all allowed logic waveforms on the rfi nodes of a supergate and the results are summed up, weighted by the probability of each case, the required supergate currents are generated.

Based on these observations, the simulation of a supergate in CREST is carried out by maintaining a set of different *simulation sub-processes*, each representing the result of a particular sequence of *logical* events at the supergate's rfi nodes. Every sub-process has a certain probability, namely the probability that the supergate has that state at that time. When a new event arrives at an rfi node it is applied to all existing sub-processes, some of which will cause the creation of new sub-processes. When two sub-processes are performing identical simulations, they are *merged* to produce a single new sub-process whose probability is the sum of their probabilities.

This approach, while acceptable for small supergates, is too expensive for larger ones. In these cases, CREST uses two heuristics to reduce the performance penalties while maintaining acceptable accuracy :

(1) Heuristic 1 : Allow only supergates whose sizes are less than a user-specified *size threshold*.

(2) Heuristic 2 : Terminate simulation sub-processes whose probability becomes less than a user-specified *probability threshold*.

## V. IMPLEMENTATION AND RESULTS

First, we present results to show the accuracy and speed of our propagation approach in circuits containing pass transistors. Fig. 3 shows results for simple pass transistor circuits using probabilistic inputs. Fig. 4 demonstrates accurate current waveforms and event placement for a 4-bit alu circuit from a recent logic design. Logical waveforms were used in this comparison since the circuit was too complex to simulate in SPICE for all the input patterns needed to generate an accurate expected current waveform. CREST simulation times for these circuits are compared with SPICE in table 1. Probability waveforms were used as the inputs for the first two test cases and they illustrate the dramatic gains available from CREST's probabilistic analysis when an expected waveform is needed. The gains possible for logical inputs are illustrated in the last two examples. Logical waveforms were used since these circuits are too large to examine for all the waveforms necessary to generate an expected waveform with SPICE.

We now present the results to demonstrate the effectiveness of the supergate simulation heuristics. Three circuits were used for these simulations: a 2-bit adder (Fig. 5), a 4-bit adder (Fig. 6), and a 4-bit multiplier (Fig. 7). The waveforms are compared in the figures indicated and the timing and speedup results of the different runs are shown in Table 2. The exact CREST simulation shown in the left half of Fig. 5 was 3000 times faster than SPICE. The right half of the figure shows a heuristic CREST run that is over four times faster with comparable current results. Fig. 6 shows the results of three different heuristic runs in CREST compared to a full-accuracy run. The first run merely eliminated extremely impropapble supergate processes and obtained 11X speedup with virtually no accuracy loss. The second run combined both heuristics and achieved excellent accuracy with 31X speedup. The final run completely eliminated supergates and shows acceptable accuracy with a 59X speedup. Fig. 7 compares a fairly tight heuristic run and a relaxed heuristic run for a 4-bit parallel multiplier. In the tight run, supergates are allowed to grow up to nine gates and supergate simulation processes are ignored only if their probability is less than .0005. In the relaxed run, the supergate size limit is set to one gate, ie "no supergates". The relaxed heuristic ran 8 times faster with comparable results. In general, the heuristics yielded comparable results with excellent increases in timing. This was particularly true for the most complex circuit, the 4-bit multiplier.

## VI. CONCLUSIONS

We have extended the CREST approach to handle general CMOS circuits, allowing pass transistors, reconvergent fanout and feedback. Two new heuristics have been presented that help maintain speed in the presence of reconvergent fanout, without significantly affecting accuracy. The results of several CREST runs on real circuits have been presented.

## VII. 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] R. Burch, F. Najm, P. Yang, and D. Hocevar, "Pattern-independent current estimation for reliability analysis of CMOS circuits", *ACM/IEEE* $25^{th}$ *Design Automation Conference*, pp. 294–299, June 12–15, 1988.

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

[4] N. P. Jouppi, "Derivation of Signal Flow Direction in MOS VLSI," *IEEE Transaction on Computer-Aided Design*, vol. CAD-6, pp. 480–490, May 1987.

[5] M. Horowitz, "Timing Models for MOS Pass Networks," *Proceedings of the IEEE International Symposium on Circuits and Systems*, 1983.

[6] 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.

**Figure 1**: Stages and wires.

**Figure 2**: CREST performance improvement as a result of the stages and wires concepts.

**(a)** **(b)**

**Figure 3**: Current results for typical pass transistor gates.

**Figure 4**: Current and voltage results for a 4-bit alu circuit.

**Table 1**: Execution time comparisons for circuits containing pass transistors. Size refers to the number of transistors.

**Figure 5**: Current waveforms for the two bit adder in Table 2.

**Figure 6**: Current waveforms for different heuristic CREST runs compared to the full-accuracy CREST run on the 4-bit ripple adder in Table 2.

**Figure 7**: Current waveforms comparisons for a fairly tight heuristic run and a relaxed heuristic run for the 4-bit multiplier in Table 2.

**Table 2**: CREST speedup results for three test circuits. The numbers under heuristic 1 (heuristic 2) column indicate the size (probability) threshold imposed on supergates. A dash indicates that the heuristic was not applied. Speedups refer to improvements in speed of each run with respect to the first run in the table for that circuit. Size refers to the number of transistors. Times are in total CPU seconds on a VAX-11/780.