# MAXIMUM CURRENT ESTIMATION IN CMOS CIRCUITS

*Harish Kriplani*[†], *Farid Najm*[††] and *Ibrahim Hajj*[†]

[†]Coordinated Science Laboratory, and
Dept. of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
1101 W. Springfield Avenue
Urbana, IL 61801.

[††]Semiconductor Process & Design Center
Texas Instruments Inc.
Dallas, TX 75265.

**Abstract:** *Excessive power supply and ground currents in integrated circuits can severely affect circuit reliability and performance. Some of the problems arising from excessive current flow are : (1) excessive voltage drop (glitches) on the power/ground lines, which can lead to soft errors, and (2) large instantaneous power dissipation, which causes overheating and ultimately leads to performance degradation. Maximum current estimates are, therefore, needed in the supply lines in order to determine the severity of these problems. These currents, however, depend on the specific input patterns that are applied to the circuit. Most previous work in this area has focused on search techniques that attempt to locate the worst case current by searching for the corresponding worst case input patterns. However, since the input space is huge, search-based algorithm for this problem can take an exponential amount of time, in the worst case. In this paper, we propose a pattern-independent, linear-time algorithm that estimates an upper bound for the Maximum Envelope Current (MEC) waveform. The MEC waveform is a point-wise maximum on all the possible waveforms that the circuit can draw. Experimental results on several benchmark circuits are provided to establish the usefulness of this approach.*

## 1 Introduction

A major concern in present day VLSI circuits is the design of supply and ground lines in a way that ensures design reliability and performance. Excessive supply currents can severely affect both circuit lifetime and performance. Some of the problems arising from excessive current flow are : (1) excessive voltage drops in the power/ground lines, and : (2) large instantaneous power dissipation. Excessive voltage drop manifests itself as *glitches* on the supply lines, and causes erroneous logic signals and soft errors. Large instantaneous power dissipation causes overheating of the devices and ultimately degrades the circuit performance. Maximum current estimates are needed in the supply lines in order to determine the severity of these problems.

Power supply and ground lines deliver power to all the gates in a circuit. The points at which the individual gates or cells are tied to the bus are called *contact points.* We need to estimate the maximum currents drawn by the circuit at every contact point in order to properly design the supply lines. A design method that uses these current waveforms to redesign supply lines for acceptable voltage drop can be found in [4].

A CMOS gate draws a pulse of current from the supply lines only when its output changes state, i.e., when it switches either low → high or high → low. For this reason, CMOS circuits pose severe difficulties for supply current estimation. We will, therefore, focus on CMOS technology. The set of excitations that are possible at each input of a gate at a particular time instant are low, high, low → high and high → low. The gate will draw different (or no) current pulses for each of these input excitations. Furthermore, the current drawn by the circuit depends also upon the time instants at which these excitations are applied. For example, if a circuit has two inputs, then the current drawn from the supply lines depends not only upon the specific excitations applied at each input but also upon the delay (or *skew*) between them.

As is clear from the above discussion, the current drawn by a CMOS circuit is a complex function of the input excitations and timing. Therefore, we need to define very carefully what we mean by the maximum current waveform at a contact point. Chowdhury and Barkatullah in [2] find the maximum of the peaks of various contact currents for all possible input patterns. In the final analysis of the supply lines, they then assume that these constant peak values are applied at the contact points for all time, i.e., they have dc currents flowing in the lines. This assumption, however, gives pessimistic results since separate sections in the circuit rarely draw their maximum currents simultaneously. In this paper, we propose a better measure of the maximum current waveform called *maximum envelope current waveform.*

The *Maximum Envelope Current (MEC) waveform* is a waveform whose value at any time $t$ is the maximum of all the current values that the circuit can draw at that time. There is a unique MEC waveform at every contact point. Let $U = \{u_1(t_1), u_2(t_2), \ldots, u_n(t_n)\}$ be an input vector that is applied to the circuit $C$, where $u_i(t_i)$ is the input excitation that is applied to the $i$th input at time instant $t_i$, and $n$ is the number of circuit inputs. Thus, the $i$th input changes state at time instant $t_i$ and $t_i$ is irrelevant if the input does not change state. Furthermore, if $[0, T]$ is the time interval of interest over which the circuit is being analyzed, then :

$$I_{MEC}(t) = \max_{\substack{u_i(t_i) \in \{low, high, lh, hl\} \\ t_i \in [0,T] \\ 1 \leq i \leq n}} I(t)$$

Here, $I_{MEC}(t)$ is the value of the MEC waveform at time $t$ and $I(t)$ is the value of the current that the circuit $C$ draws from the supply lines at time $t$ when the input vector $U$ is applied to it. The abbreviations $lh$ and $hl$ stand for low → high and high → low transitions, respectively. Thus, $I_{MEC}(t)$ is the maximum possible current value that could be drawn from the supply lines at time $t$, given that each input can switch at anytime during the interval [0, $T$]. We are interested in algorithms that efficiently estimate the MEC waveform.

Accurate estimation of the MEC waveform at every contact point is extremely difficult since we need to determine current waveforms corresponding to all possible input patterns. If the circuit has $n$ primary inputs then we need to explore the set of $4^n$ input patterns to calculate the MEC waveforms. Moreover, as was pointed out before, each component of the input vector can be applied to the circuit at any time during the interval $[0, T]$. This compounds the problem and makes it practically impossible to handle by any of the known search procedures. As will be shown in the next section, most previous work in this area has been based on search techniques. In this paper, we propose linear time (in the number of gates) algorithms that provide tight upper bounds on MEC waveforms. The proposed approach represents a trade-off between execution speed and tightness of these bounds.

This paper is organized as follows. In the next section, we briefly discuss previous and related work in this area. In section 3, we discuss various assumptions on which our approach is based. In section 4, we present the algorithm in detail. Experimental results are presented in section 5. In order to maintain reasonable execution times, our algorithm neglects signal correlations that exist inside the circuit. As mentioned above, the price one has to pay for this is looser upper bounds on the MEC. The signal correlation problem is described in section 6. In section 7, we present a heuristic technique that partially accounts for signal correlations in order to obtain tighter bounds. Finally, in section 8, conclusions and some guidelines for future work are presented.

## 2 Previous Work

Chowdhury and Barkatullah have addressed the problem of maximum current estimation in [2]. In their methodology, they divide the circuit into a set of macros, each of which consists of a combinational interconnection of logic gates. Considering each macro separately, they use either a (branch and bound) search technique or a heuristic technique to find the maximum of its transient current, assuming its inputs switch simultaneously. However, they do not discuss the effect of the various macro interconnections on the overall maximum current. In addition, they assume that internal nodes make single transitions. Our experience with various benchmark circuits indicates that the contribution of multiple node transitions to the supply currents can be significant. Due to the huge size of the input space, their branch and bound search technique is slow on large circuits. Furthermore, their heuristic approach does not guarantee an upper bound on the maximum currents.

**Figure 1.** An example of a latch controlled synchronous digital circuit.

Devadas et. al. have addressed a similar problem in [3]. They consider the estimation of worst case power dissipation in CMOS combinational circuits. They reduce this problem to a weighted max-satisfiability problem, on a set of multi-output Boolean functions obtained from the circuit logic description. These functions are appropriately weighted to account for the different load capacitances. A branch and bound algorithm is then used to solve the ($\mathcal{NP}$-complete) max-satisfiability problem. They are able to account for multiple node transitions. However, for a multi-level logic circuit, even under a unit gate delay assumption, the functions generated by their algorithm are fairly complex. Consequently, even for small circuits, their analysis is slow. Analysis of multi-level circuits under a general delay model was not attempted.

¿From this brief survey, it is clear that existing methods for the calculation of maximum current are computationally too expensive to handle large VLSI circuits. In order to be able to handle present and next generation of VLSI circuits, linear time algorithms are necessary. In section 4, we present a linear time algorithm for the MEC waveform estimation. But first, we discuss various simplifying assumptions that will be used in that algorithm.

## 3 Assumptions

In order to reduce the complexity of the problem, we focus on a specific, but very common design style, namely (edge-triggered) latch-controlled synchronous digital circuits. These circuits consist of combinational blocks separated by latches (see Figure 1) such that all inputs to each combinational block switch simultaneously. As a result, we will be able to focus the analysis in the next section on a single combinational block whose inputs all switch at time 0. This effectively eliminates the time domain uncertainty about the input transitions, and significantly simplifies the problem. This assumption has also been used by both previous approaches surveyed above, although it has not always been clearly mentioned.

We assume that the delay of each gate in the circuit is fixed and is specified ahead of time. But different gates can have different delays. Further, we assume that each time the output of a gate switches, a triangular pulse of current is drawn from the supply lines, as shown in Figure 2. The duration of this pulse is

**Figure 2.** Model of a gate current pulse.

computed from the delay of the gate (by charge conservation), and its height (peak value) is a user-specified value. Two separate values for the peak current are allowed, corresponding to *lh* and *hl* transitions at the gate output.

Given the specific clocking scheme of the synchronous circuit, the MEC waveforms of combinational blocks driven by the same clock can be grouped together and used to find the bus branch currents. For purposes of this paper, we will focus (in the next section) on the analysis of a single combinational block, tied to the bus at a single contact point.

## 4 Description of the Proposed Algorithm (iMax)

Given a gate level description, we assume that little or nothing is known about the values at the primary inputs, except that they may only transition at time zero. We call this an *uncertainty* about these input signals. The basic idea is to propagate this uncertainty into the circuit, so that, at every logic gate, we know the range of possible values that its inputs can take. From this, the worst case gate current is computed, as explained below.

Perhaps the first question to come to mind is as to the kind of information that one maintains in order to represent the uncertainty about internal circuit nodes. Ideally, one would like to compute the set of *all* possible transitions (along with their timing information) that occur at every gate in the circuit. That would certainly be enough to estimate the MEC waveform. However, given the uncertainty at the inputs, the number of possible transitions at internal gates grows exponentially, and quickly becomes a bottleneck. To avoid this problem we maintain information, not about individual transitions, but about *intervals* during which the output of the gate *might* switch. Thus, for each of the excitations *low*, *high*, *hl* and *lh*, we store a list of intervals during which a node might carry that excitation. These intervals, which might overlap, serve to describe the signal uncertainty.

An example of this signal representation is given in Figure 3. In that figure, we show an uncertain signal $X(t)$ represented as four sets of intervals along the time axis. Thus, if $x(t)$ is a logic signal that belongs to the family $X(t)$, i.e., $x(t) \in X(t)$, then $x(t)$ is low up to $t_1$, switches from low to high sometime between $t_1$ and $t_2$, is then high up to $t_3$, etc. Thus at any time between $t_1$ and $t_2$ the signal can be either high or low. Notice that between $t_6$ and $t_7$ the signal may make any number of low to high and high to low transitions. At the primary inputs, the signals are represented by such waveforms with a single point of *possible* transition at time 0. As internal signals are generated, the number of possible transition points increases. In order to contain the complexity, we then start to merge neighboring transition points into intervals. In general, this strategy can be stated as follows : when the number of intervals associated with a gate corresponding to any excitation exceeds a certain user-specified threshold (Max_No_Hops), we repeatedly merge closest-neighbor intervals, so as to keep their number down.

**Figure 3.** An illustration of signal uncertainty at a gate output.

Given such waveform information at the inputs of a logic gate, we would like to derive the corresponding information about its output. One cannot do this accurately, though, without knowing how some of these inputs, if any, are *correlated.* For instance, certain *combinations* of the gate input values may not be possible. Unfortunately, maintaining information on the correlation between various circuit nodes is too expensive. We, therefore, use a *conservative* approximation, one that does *not* underestimate the MEC waveform, as follows. If we assume that *all* combinations of the gate input values are possible, i.e., that the gate inputs are *independent*, then the worst case current in that case *is* an upper-bound on the gate current for that range of inputs. In other words, the worst case current over *all* combinations of inputs is certainly an upper bound on the worst case current over *some.*

Given the type of a Boolean gate, and using the independence assumption, it is a simple matter to compute the intervals at the gate output from those at its inputs (see Figure 4). ¿From this, the supply current contribution of the gate is also calculated. For instance, if a gate can switch at any time during a certain interval, then the waveform shown in Figure 5 is considered to be its contribution to the supply current for that interval. This waveform is an envelope of all possible currents that the gate can draw due to a transition occurring at any time during that interval. Thus, it represents the worst case scenario. At every gate, there are two possible current waveforms, one due to low → high transitions (LHCurrent) and the other due to high → low transitions (HLCurrent). The maximum of the two current waveforms at every time point is considered to be the worst case supply current. Once all the gate currents have been computed, the current waveform at the contact point is determined by adding together the individual gate contributions.

**Figure 4.** An example of output intervals computation.

**Figure 5.** Current waveform due to a transition interval.

The above approach has been implemented in a program called **iMax**. In the program, the circuit is first *levelized* so that a gate at level $j$ does not feed any other gate at a level less than or equal to $j$. Any user-specified restrictions on certain inputs are then imposed, while all other inputs are assumed to take all possible values from the set $\{$ *low*, *high*, *lh*, *hl* $\}$. After this, the circuit is analyzed in a level by level fashion, starting from the lowest level, propagating the uncertainty at the inputs of every gate to its output. As a result, we get a current waveform that is a point-wise upper bound on the MEC waveform.

In order to assess the quality of the solution, we need to determine how tight the upper bound obtained by iMax is. One way of doing this would be to perform an exhaustive enumeration over all possible input patterns and actually calculate the MEC. However, this would be very expensive and practically impossible for circuits with more than 9-10 inputs (Note: $4^{10} = 1,048,576$). We, therefore, resort to the following random enumeration approach. We repeatedly apply input patterns to the circuit, randomly selected from the set of all possible patterns, and use an event-driven logic simulator to calculate the possible excitations at the outputs of various gates. From these excitations, supply current waveforms are easily calculated. By maintaining the upper-bound envelope of all these waveforms, we basically get a lower-bound on the MEC. Naturally, the more patterns are simulated the closer this waveform will be to the MEC. Ideally, one would like to see the upper-bound obtained from iMax come as close to this lower-bound as possible. The program that implements this random enumeration technique is called **iLogSim** (Current Logic Simulator).

## 5 Experimental Results

In all the circuit examples considered below, two assumptions are made. First, a constant number is assigned to every gate as its delay value. This delay value can be different for different gates. Second, the peak of the transition current for every gate for both *lh* and *hl* transitions is (arbitrarily) taken to be 2 units of current. The results of iLogSim on various circuits were collected after running the logic simulator 10,000 times with random input patterns. As mentioned earlier, we also assume that all the gates are connected to a single contact point, and we report the peak values of the current waveforms obtained from iMax and iLogSim.

Table 1 lists the results of running iMax and iLogSim on nine small CMOS circuits. The number (10) next to iMax in the table indicates the value of Max_No_Hops. In most of the cases, the results of iMax are in perfect agreement with the iLogSim results. In Table 2, we report the results of running the algorithms on the ten ISCAS-85 benchmark circuits [1].

We observe that for all the circuits, the linear time iMax algorithm took only a few seconds of cpu time on a sun SPARC station 2. Further, for most of these circuits, the upper bound obtained from iMax is within 80% of the lower bound obtained from iLogSim. There are two possible reasons for this mismatch. Firstly, it is quite possible that the lower bound obtained from iLogSim is not very close to the MEC waveform. Since all the circuits have at least 32 inputs, the space of possible input patterns is huge, and one should not expect that the waveform based on 10,000 input patterns will be very close to the MEC. For smaller circuits (Table 1) where the input space is not so huge, 10,000 input patterns were enough to get a lower bound estimate of MEC waveform that is quite close to the actual waveform, as is reflected by the results. The second possible source of mismatch is our conservative independence assumption for the signals at the gate inputs. One can improve on this assumption by attempting to resolve the signal correlations, as discussed in the following sections.

We next discuss the effect of varying the Max_No_Hops parameter on the performance of iMax. Table 3 lists the results obtained from iMax on ISCAS-85 circuits.

¿From the table, it can be seen that the peak of the total current waveform improves as Max_No_Hops increases, though it does not improve significantly beyond Max_No_Hops = 5. As the value of this parameter is increased, the amount of memory being used by the program increases, and the number of intervals being merged decreases. A value between 5 and 10 seems to be a good choice for this parameter.

## 6 Signal Correlations

In general, signal values at internal nodes are correlated, which limits the number of transitions that can occur at the output of a gate. The source of the correlation problem is situations where the output of a gate fans out to several other gates. Such gates are called *multiple fan-out gates.* While analyzing each of the gates that are directly connected to a multiple fan-out gate, we should assign the same value to all the lines connected to its output. Thus, the outputs of each of the gates that are directly connected to a multiple fan-out gate are correlated. In other words, even though each of these gates can assume all possible transition values calculated by iMax, they may not *simultaneously* carry their worst case excitations. As one goes deeper into the circuit, where these correlated outputs reconverge and feed the same gate, the inputs of that gate become correlated. Such gates are called *reconvergent fan-out gates.* With correlated inputs, the number of transitions that can possibly occur at the output of the gate is reduced. In iMax, while analyzing a gate, we assume that each of its inputs can take all possible values from the set of possible values as calculated by iMax, i.e., we assume that the gate input lines are independent. Thus, the basic iMax algorithm described above completely ignores all signal correlations and, therefore, overestimates the supply currents.

The advantage of ignoring correlations in the basic iMax algorithm is its, very desirable, linear time performance. In the next section, we propose a heuristic that accounts for signal correlations, at the expense of increased execution time.

## 7 Multi-Cone Analysis (MCA)

One can always compute the MEC waveform by brute-force input enumeration. By the same token, we can *improve* on the result of the basic iMax algorithm by doing *partial enumeration.* By enumerating the possible signal values at only a small subset of the circuit nodes, we achieve a trade-off between accuracy and speed. Because of their influence on signal correlation, it turns out that the best candidates for partial enumeration are the multiple fan-out nodes. By enumerating all possible values at a fan-out node and estimating the total circuit current corresponding to each, one eliminates the signal correlations arising from it and obtains a better upper bound on the MEC.

To illustrate these ideas, let $N$ be a multiple fan-out node at the primary inputs. Let $X(t)$ be an *ambiguous* or uncertain waveform at $N$, as described in section 4. Thus $X(t)$ is a set of possible waveforms $x(t) \in X(t)$, each of which is a possible logic signal at that node. Let $t_0$ be a fixed time instant and define $X_l(t)$ to be the set of all $x(t) \in X(t)$ for which $x(t_0)$ is low. Likewise, define $X_h(t)$, $X_{lh}(t)$, and $X_{hl}(t)$, based on $x(t_0)$. These sets form a disjoint partition of $X(t)$. Let $Y_i, i = 1, \ldots, m$, be the uncertain waveforms at all *other* primary inputs. If $I_{MEC}(t)$ is the MEC waveform and $I(t)$ is the current corresponding to a specific selection of $x(t) \in X(t)$ and $y_i(t) \in Y_i(t)$, then :

$$I_{MEC}(t) \leq \max_{\substack{x(t)\in X(t)\\ y_i(t)\in Y_i(t)}} I(t) = \max \left\{ \max_{\substack{x(t)\in X_l(t)\\ y_i(t)\in Y_i(t)}} I(t), \right.$$

$$\left. \max_{\substack{x(t)\in X_h(t)\\ y_i(t)\in Y_i(t)}} I(t), \max_{\substack{x(t)\in X_{lh}(t)\\ y_i(t)\in Y_i(t)}} I(t), \max_{\substack{x(t)\in X_{hl}(t)\\ y_i(t)\in Y_i(t)}} I(t) \right\}$$

Here equality holds when all $Y_i(t)$'s are uncorrelated.

By restricting $X(t)$ so that $x(t_0)$ is low, we can use iMax to obtain an upper bound on $\max_{\substack{x(t)\in X_l(t)\\ y_i(t)\in Y_i(t)}} I(t)$, call it $I_l(t)$. We can likewise compute the upper bounds $I_h(t)$, $I_{lh}(t)$, and $I_{hl}(t)$. From the above, it's easy to see that :

$$I_{MEC}(t) \leq \max\{I_l(t), I_h(t), I_{lh}(t), I_{hl}(t)\}$$

Thus, by enumerating the possible values of $x(t_0)$, running iMax for each case and taking the maximum, we obtain an upper bound on the MEC waveform. Since in each such run of iMax the fanout branches of $N$ take *specific* known values, correlation between them at $t_0$ is no longer an issue, and the resulting upper bound on MEC should be an improvement on the original upper bound. This analysis is true for any $t_0$. Therefore, if we select a number of candidate $t_0$ values and do this enumeration at each, the *minimum* of the upper bounds obtained in each case can be chosen to be the best available upper bound on the MEC waveform. In our implementation, we choose $t_0$ to be at the beginning and at the end of each of the intervals in $X(t)$.

Ideally, one would like to enumerate all possible *combinations* of values at the multiple fan-out nodes. However, this becomes too expensive since there are typically many multiple fan-out lines in a circuit. We, therefore, enumerate every fan-out node on its own. The overall procedure is as follows. We first run the basic iMax algorithm on the circuit and determine the set of (unconstrained) transitions that can occur at every gate. In a second pass, we separately examine each multiple fan-out line and enumerate its values at a number of discrete time points $t_0$.

While enumerating a particular multiple fan-out line, we do not need to recompute the signal transitions at *all* the gates in the circuit. It is enough to process only those gates that lie in the COne of INfluence (COIN) of the fan-out line. The COIN of a line consists of all the gates whose outputs are affected by a change in excitation at the line. Thus, a gate is in the COIN of a line if it is either directly connected to it or connected to the output of a gate which is in the COIN. The COIN can be constructed by a simple depth-first traversal starting at the fan-out line. The *distance* of a gate from a line is defined as the minimum number of gates present on any path between the gate and the line. A fan-out node has little direct influence on gates that are very far removed from it. It, therefore, makes little sense to maintain very deep COINs. Based on this, in a COIN, we only include those gates whose distance from the parent fan-out line is less than or equal to a user-specified parameter Depth_FO_Cone.

While a COIN is being analyzed, we consider that the remainder of the circuit is unconstrained, and draws its full-range of possible currents as initially computed by iMax. As a result, irrespective of the choice of COINs and of the order in which they are enumerated, our partial enumeration algorithm always gives an upper bound on the MEC.

In table 4, we report the results of applying the above algorithm (MCA) to the ISCAS benchmark circuits. For most of the circuits, this leads to about 7–8% improvement in results at a small increase in execution time.

## 8 Conclusions

We have proposed a linear time algorithm that computes the maximum currents in the supply lines. Most of the previous algorithms on maximum current estimation suffer from exponential complexity and are not adequate for large circuits. Our approach avoids the exponential complexity by adopting a pattern independent approach. The algorithm presented is able to find an upper bound on MEC waveform in circuits with 100's of inputs and 1000's of gates in a few minutes of cpu time on a sparc SUN workstation. For circuits with few inputs, where it is possible to explore the input search space and determine what the actual MEC waveform is, our algorithm produced extremely good results. For most of the larger circuits, the upper bounds produced by our algorithm are within 60% of the lower bounds obtained by exploring a very small portion of the input space. We are currently conducting further research on the signal correlation problem.

## Acknowledgements

A major part of this research was performed while the first author was at Texas Instruments for the Summer of 1991. The authors are thankful to the technical staff members of the Semiconductor Process and Design Center at TI Dallas, especially Dr. Ping Yang and Dr. Jue-Hsien Chern, for providing valuable discussions and guidance. This research is supported by Texas Instruments Inc. and the Semiconductor Research Corporation.

## References

[1] F. Brglez and H. Fujiwara, "A Neutral Netlist of 10 Combinational Benchmark Circuits and a Target Translator in Fortran," *Proc. of International Symposium on Circuits and Systems*, 663–698, June 1985.

[2] S. Chowdhury and J. S. Barkatullah, "Estimation of Maximum Currents in MOS IC Logic Circuits," *IEEE Trans. on Computer-Aided Design*, Vol. 9, No. 6, pp. 642–654, June 1990.

[3] S. Devadas, K. Keutzer and J. White, "Estimation of Power Dissipation in CMOS Combinational Circuits," *IEEE Custom Integrated Circuits Conference*, pp. 19.7.1–19.7.6, 1990.

[4] J. E. Hall, D. E. Hocevar, P. Yang and M. J. McGraw, "SPIDER – A CAD System for Modeling VLSI Metallization Patterns," *IEEE Trans. on Computer-Aided Design*, Vol. CAD-6, No. 6, pp. 1023-1031, November 1987.