Extreme Delay Sensitivity and the Worst-Case Switching Activity in VLSI Circuits[†]

Farid N. Najm and Michael Y. Zhang

ECE Dept. and Coordinated Science Lab.
University of Illinois at Urbana-Champaign
Urbana, IL 61801

Abstract – We observe that the switching activity at a circuit node, also called the *transition density*, can be *extremely sensitive* to the circuit internal delays. As a result, slight delay variations can lead to *several orders of magnitude* changes in the node activity. This has important implications for CAD in that, if the transition density is estimated by simulation, then minor inaccuracies in the timing models can lead to very large errors in the estimated activity. As a solution, we propose an efficient technique for estimating an *upper bound* on the transition density at every node. While it is not always very tight, the upper bound is *robust*, in the sense that it is valid irrespective of delay variations and modeling errors. We will describe the technique and present experimental results based on a prototype implementation.

I. Introduction

Higher levels of integration and shrinking line widths have led to a generation of devices that have more severe power dissipation and reliability problems than typical devices of a few years ago. Excessive power dissipation may cause run-time errors and device destruction due to overheating, while reliability issues may shorten device lifespan. It is especially useful to diagnose and correct these problems before circuits are fabricated. In the popular CMOS technology, logic gates draw current and consume power only when making logical transitions. As a result, power dissipation and reliability strongly depend on the extent of circuit *switching activity.*

Circuit activity is dependent on the input patterns being applied to the circuit. For one input set the circuit may experience no transitions, while for another it may switch very frequently. During the first input set the circuit dissipates little power and experiences little wear, but for the second its activity might cause device failure. Thus one is tempted to simulate the circuit for all possible inputs in order to measure the activity, which is highly impractical for VLSI.

Recently, some approaches have been proposed to solve this problem by using probabilities to represent *typical* behavior at the circuit inputs. In [1], the average number of transitions per second at a circuit node is proposed as a measure of switching activity, called the *transition density.* An algorithm was also proposed to propagate specified input transition densities into the circuit to compute the densities at all the nodes. Other approaches, such as [2] and [3], have also been proposed to overcome the pattern dependence problem and estimate the transition density.

However, in addition to being input pattern dependent, the transition density at a node also depends on the path delays inside the circuit. Thus, due to different path delays, a node in a clocked synchronous circuit may make several transitions before settling down to its steady state value in a clock period. Indeed, as we will illustrate in the next section, the transition density at a node can be *extremely sensitive* to the circuit internal delays. As a result, slight delay variations (due, say, to imperfections in the manufacturing process) can lead to *several orders of magnitude* changes in the switching activity. Furthermore, if the transition density is estimated by simulation, such as in [3], then minor inaccuracies in the delay models can lead to large errors in the estimated activity. Likewise, both approaches [1] and [2] can develop accuracy problems resulting from extreme sensitivity. In [1], the circuit delays are not explicitly taken into account, so the error due to delay sensitivity becomes part of the overall approximation error of the technique. In [2], the symbolic expressions representing the probability of switching depend explicitly on the circuit delays and will, therefore, have accuracy problems due to delay sensitivity.

To address this problem, we propose a method of estimating an *upper bound* on the transition density of individual nodes within a combinational circuit (assumed to be embedded in a larger sequential circuit). The upper bound provides an estimate of the *maximum transition density* at

[†] This work was supported in part by the National Science Foundation (NSF), under grant MIP-9308426.

a node, and is based on user-specified min-max delay intervals for each logic gate. This estimate is *robust*, in the sense that it is valid irrespective of delay variations and timing model inaccuracies (within the specified delay intervals). The technique uses signal *uncertainty* to capture the worst case behavior of the circuit. It has been implemented in a prototype simulator, called MaDest (Maximum Density Estimator).

The rest of this paper is organized as follows. In the next section, the notion of extreme sensitivity is illustrated in more detail. Section III contains a detailed description of the upper bound computation algorithm, and experimental results are presented in section IV. Finally, a summary and conclusion are presented in section V.

II. Extreme Sensitivity

We will illustrate the extreme sensitivity phenomenon with the help of the two circuits in Figs. 1 and 2. The circuit in Fig. 1 (Circuit A) was simulated using [3] to compute the transition density at every node to within 1%, with 95% confidence. All inputs were assigned a (normalized) transition density of 0.5 and a probability of 0.5. This means that, on average, a primary input makes a transition every other clock cycle, and spends half the time in the logic 1 state. Another circuit, shown in Fig. 2 (Circuit B) was obtained from circuit A by simply removing the NOR gate *q*. This circuit was then also simulated using [3] under the same conditions. As far as the output node 32 is concerned, the only difference between the two circuits is the slight change in the delay of the path (10, 22, 32) due to the reduced capacitive loading when the NOR gate is removed. This slight change in one of the path delays causes the transition density at the output node to vary by a factor of 0.38/0.0005 = 760, almost three orders of magnitude, between the two circuits. Thus node 32 is said to be *extremely sensitive.*

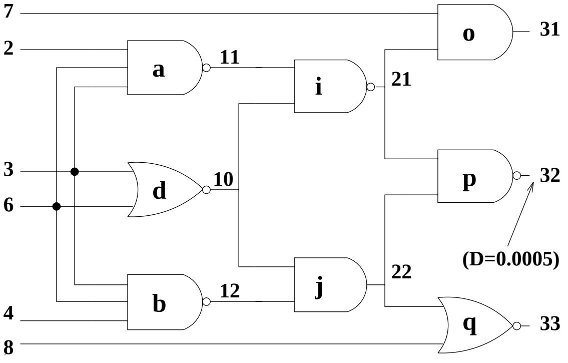

Figure 1. Circuit A - node 32 has very low transition density.

The implication for CAD is profound: if the transition density is estimated by simulation, then minor inaccuracies in the simulation models can lead to large errors in the estimated activity. Even if circuit simulation were used to estimate switching activity, which would be prohibitively expensive, extreme sensitivity may still be a problem. This is because slight delay variations due to imperfections in the manufacturing process may still lead to order of magnitude changes in the switching activity. To overcome this problem, we will propose in the next section an efficient technique for computing an *upper bound* on the transition density that is valid irrespective of delay variations or modeling errors. Using this technique, the user is alerted to the possibility of having very high transition density at some nodes. More detailed analysis can then be carried out on these nodes, and corrective design measures can be implemented. Before going on, however, we will make some observations regarding the cause of the extreme sensitivity.

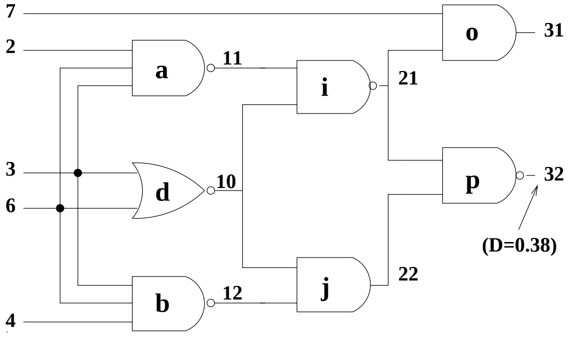

Figure 2. Circuit B - node 32 has much higher transition density.

Circuits A and B are not unique. Indeed it turns out that a *necessary* condition for a node to be extremely sensitive is that it be located where two or more *reconvergent paths* meet, provided the paths have delays that differ by a small amount, approximately equal to the inertial delay of the gate. Slight delay variations can then have a large impact on the transition count, because if the difference in path delays becomes less than the inertial delay, events will cancel out and few output events will be generated. Otherwise, if the difference in path delays is larger than the inertial delay, then multiple events may be generated at the gate output. This condition is not *sufficient*, however, for extreme sensitivity. The signal values and the Boolean properties of the paths play an additional key role in determining extreme sensitivity. For instance, two competing events at the inputs of an AND gate must be complementary, otherwise a single event will be generated irrespective of the delays.

Based on this necessary condition, we have implemented a simple pre-processor that examines the circuit topology and flags a node as *potentially extremely sensitive* if it satisfies the necessary condition. The results indicate that a low percentage of nodes are potentially extremely sensitive (3.2% for the ISCAS-85 benchmarks). Although this is a small fraction of nodes, one cannot ignore this problem because if one of these nodes is close to the primary inputs, then its density value will affect all other nodes downstream from it, leading to large variations in the estimated circuit activity.

III. Upper Bound Computation

We assume that the circuit to be analyzed is a combinational block that is part of a larger synchronous sequential design, as shown in Fig. 3.

The primary inputs to the circuit (combinational block) switch in synchrony with the clock, if at all, and can make at most one transition per clock cycle. Other circuit nodes, however, can make multiple transitions per clock cycle. Let $n_x(T_c)$ denote the number of transitions at node x in one clock cycle T_c. For a given node, the *average* (or *expected*) number of transitions per clock cycle, divided by the clock period, is the *transition density* for that node [1] :

$$D(x) = \frac{E[n_x(T_c)]}{T_c}$$

If $\hat{n}_x(Tc)$ is the *maximum* possible number of transitions per clock cycle at node x, then $\hat{n}_x(T_c)/T_c$ is the *maximum transition density*, so that:

$$D(x) \leq \frac{\hat{n}_x(T_c)}{T_c}$$

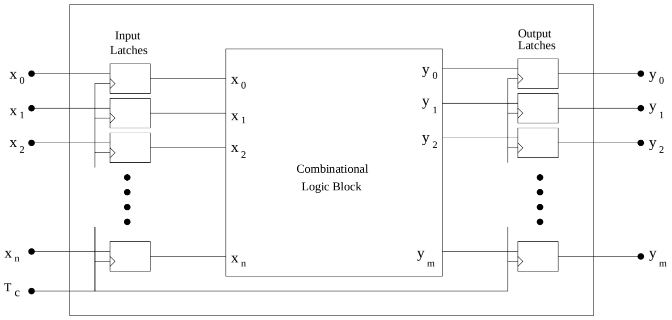

Figure 3. A combinational circuit embedded in a synchronous sequential design.

We propose a method of computing an upper bound on $\hat{n}_x(T_c)$ that is independent of the sensitivities and delay variations. We denote this upper bound by $U[n_x(T_c)]$. Effectively, this leads to an *upper bound on the transition density* :

$$D(x) \leq \frac{U[n_x(T_c)]}{T_c}$$

Ideally, we would like $U[n_x(T_c)]$ to be equal to $\hat{n}_x(T_c)$. However, in order to maintain computational efficiency, we can not guarantee this and, in general, they will not be equal.

A. Signal uncertainty representation

We will represent the variety of possible waveforms at a circuit node with a single waveform that helps describe our *uncertainty* about the behavior of the real signal. An example of this representation is given in Fig. 4.

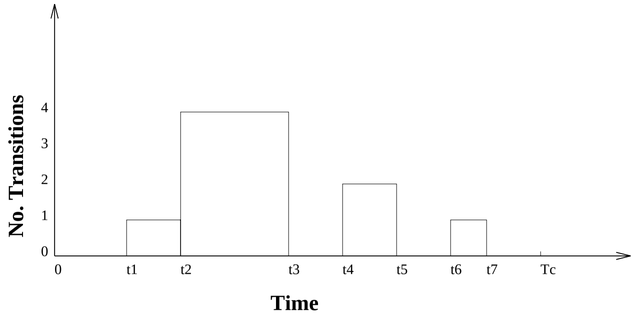

Figure 4. Signal uncertainty representation.

The vertical axis gives the maximum number of transitions (integer valued) that a node can possibly experience in specified time intervals. Thus the waveform in Fig. 4 indicates that this node makes *at most* 1 transition between t_1 and t_2, *at most* 4 transitions between t_2 and t_3, *at most* 2 transitions between t_4 and t_5, *at most* 1 transition between t_6 and t_7, and no transitions at any other time within the clock period T_c.

Once such a waveform is available for every circuit node, then adding the transition count associated with each interval gives the required upper bound at that node, $U[n_x(T_c)]$. We derive the signal uncertainty waveform associated with each circuit node by propagating user-specified uncertainty waveforms from the primary inputs throughout the circuit. A primary input node can have at most one transition. Thus the corresponding waveform consists of a single interval in which the transition count is 1. Ideally, this interval consists of the single time point $t = 0$. However, in order to allow for clock skew or delay variations, we use a more general model in which this interval is specified as $[0, t]$ where t is under user control (or a program default), as shown in Fig. 5.

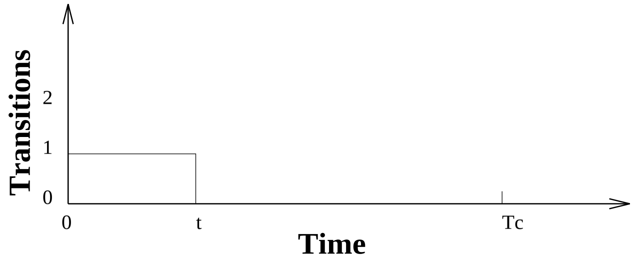

Figure 5. The signal uncertainty representation for a primary input node.

B. The propagation algorithm and heuristic

The propagation algorithm visits every gate in the circuit only once, starting at the primary inputs, and processes a gate only when all its fan-in gates have been processed. It is assumed that the gate delays are not known exactly, but are only known to be within user-specified intervals $[t_{d,min}, t_{d,max}]$. This allows for delay variations due to process variations, temperature, drift, and timing model inaccuracies. The delay limits, which may be different for different gate types, are scaled by the fanout capacitance seen by the gate.

When processing a gate, the uncertainty waveforms at its inputs are examined, and a corresponding uncertainty waveform is generated at its output. Since the logic values and specific transition times at the gate inputs are not known, the only way to *guarantee* an upper bound on the output transition count is to assume that every input transition goes through. In this case, the output transition count of a gate is simply the sum of all its inputs' transition counts. To illustrate, consider an AND gate with inputs A and B, and output C, for which the input and output uncertainty waveforms are shown in Fig. 6. Notice that the time interval at the output node is *expanded* to allow for maximum and minimum propagation delays.

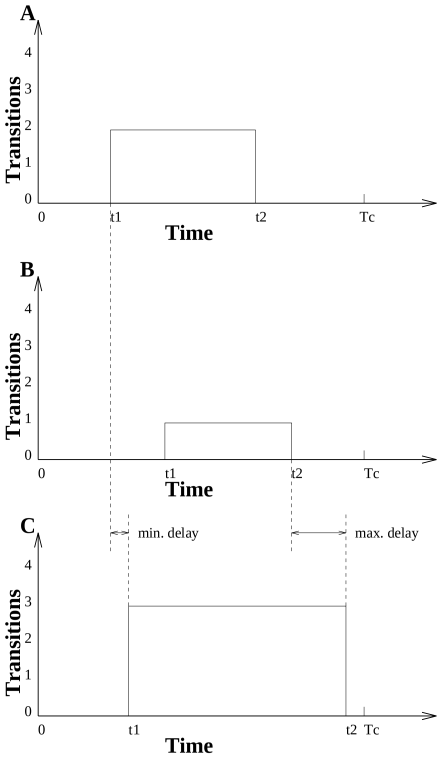

Figure 6. Input and output transition characteristics.

It should be clear that this simple propagation procedure is very fast, but can lead to loose upper bounds. While this is true in general, we have found that by using two modifications to the basic technique, we can achieve reasonable accuracy without impairing the speed advantage of the approach.

The first modification has to do with the fact that logic gates have non-zero *inertial delay.* Thus the output of a logic gate cannot carry arbitrarily short pulses. A pulse has to be at least as long as the inertial delay if it is to be transmitted. Therefore, every output node has a *minimum pulse width* that puts a ceiling on the number of transitions that it can have within each sub-interval.

The second modification is a heuristic that we have found works well in practice, as shown in section IV, and which tries to account for the *other* gate inputs. One reason that the upper bound can be loose is that whether or not a transition propagates through a gate depends on the signals at the other gate inputs. Among logic gates, only the exclusive-or (XOR) gate has no *controlling* input value, and thus allows more transitions to go through. All other gate types (NAND, AND, NOR, and OR) will block some transitions when one of their inputs is at a controlling value (0 for AND and NAND, 1 for NOR and OR). To represent this fact, it seems reasonable to compute the maximum transition count for a logic gate (other than XOR) as some *fraction* of the sum of its input transitions, rather than the whole sum. We have found that a fraction of 3/4 works well in practice. To see why this factor is plausible, consider that, for a 2-input XOR gate, there are 4 combinations of input transitions that produce an output transition, as illustrated by the solid lines in Fig. 7a. In contrast, only 3 combinations of input transitions produce an output transition in the case of an AND gate, as shown in Fig. 7b. Hence the 3/4 factor.

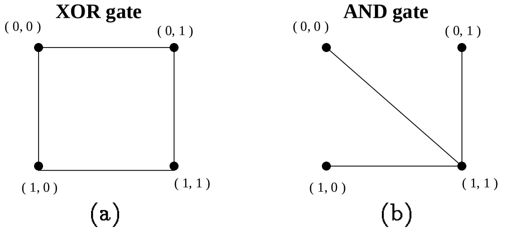

Figure 7. Transition diagrams for (a) an XOR gate and (b) an AND gate.

Similar analysis yields the same 3/4 factor for every other gate type (other than XOR) and for any number of inputs. The experimental data presented in the next section demonstrate that this works well in practice.

IV. Experimental Results

The proposed technique has been implemented in the prototype program MaDest. The program uses a simplified gate timing model in which library-specified propagation delays are scaled by the external capacitive loading. The gate library also specifies min-max delay intervals for every gate. We will present results for the ISCAS-85 benchmark circuits [4] (after mapping to the gate library). In order to study the accuracy, one needs the true "maximum number of transitions per clock cycle" at every node, allowing for timing model inaccuracies, process and delay variations, clock skew, etc. Finding this would be extremely computationally expensive. Instead, we will compare the upper bound densities to the *maximum observed* densities obtained from very long logic simulations, using [3]. It should be clear that the true maximum should be at least as high as that observed from any simulation run. Thus the *maximum observed* densities measured from simulation are actually *lower bounds* on the true maximum.

As a result, all accuracy comparisons will be made between the upper bounds produced by MaDest and the lower bounds obtained from simulation. Thus the error measurements presented below will be worst case, i.e., upper bounds on the true errors. We will use the following error measures to study the accuracy, where $n_x(T_c)$ is the number of transitions at node x in one clock cycle, $U[n_x(T_c)]$ is the computed upper bound, $L[n_x(T_c)]$ is the lower bound obtained from [3], and N is the total number of nodes in the circuit:

$$\text{Relative Error}(n_x) = \frac{U[n_x(T_c)] - L[n_x(T_c)]}{L[n_x(T_c)]}$$

$$\bar{\epsilon} = \text{Average Relative Error} = \frac{1}{N}\sum_{x=1}^{N} \text{Relative Error}(n_x)$$

Table I.
Average Relative Error for the ISCAS-85 Benchmark Circuits.

Circuits	#levels	#gates	$\bar{\epsilon}$	$\bar{\epsilon}_h$
c432	17	160	39.4%	7.4%
c499	11	202	64.7%	44.2%
c880	24	383	73.0%	20.7%
c1355	24	546	155.2%	57.8%
c1908	40	880	68.1%	32.6%
c2670	32	1193	60.3%	36.7%
c3540	47	1669	96.7%	63.2%
c5315	49	2307	60.2%	26.3%
c6288	124	2416	180.7%	131.0%
c7552	43	3512	93.1%	49.4%

The *average relative errors* observed for the ISCAS-85 circuits are shown in Table I, both with ($\bar{\epsilon}_h$) and without ($\bar{\epsilon}$) the heuristic. The heuristic works well and leads to considerable improvement in the upper bound. On average, the density results are overestimated by a factor of about 1.5. To investigate this further, consider the histogram of the relative errors at all the nodes in c432, shown in Fig. 8. While most nodes have low error values, the errors for a few of the nodes is quite high. Undoubtedly, this is due in part to the approximations that have been used, and suggests that one should try to do better. However, in some cases the high error is simply a result of the node sensitivities. For instance, we have verified that the three nodes with the largest error values in Fig. 8 are *potentially extremely sensitive* - they satisfy the necessary condition of section II. Similar trends were observed in other circuits: nodes that are potentially extremely sensitive typically exhibit large relative errors. Thus, while it is not always so, in many cases the "error" observed is not so much an accuracy problem as it is simply an indication of the presence of extremely sensitivity nodes.

It should be clear that the approach is very fast, and has a time complexity that is linear in circuit size. Indeed, the execution time (SUN Sparc ELC workstation) was under 1/2 second for any one ISCAS-85 circuit. The largest circuit took only 0.34 cpu seconds.

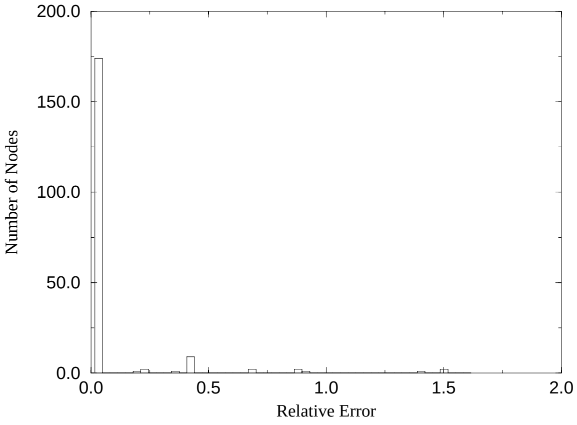

Figure 8. c432 transition density relative error histogram with the heuristic.

VI. Summary and Conclusion

The average number of transitions per second at a circuit node is a measure of switching activity called the *transition density*. We have observed that in some cases, the transition density at a node can be extremely sensitive to the circuit internal delays. As a result, delay variations due to process imperfections can lead to order of magnitude changes in the switching activity. Furthermore, if the transition density is estimated by simulation, then minor inaccuracies in the delay models can lead to large errors in the estimated activity.

As a solution, we have proposed an efficient technique for estimating an upper bound on the transition density at every node. The upper bound is *robust*, in the sense that it is valid irrespective of delay variations. Experimental results demonstrate that the technique is fast, and that a simple heuristic can be used to significantly improve the tightness of the bound.

References

[1] F. Najm, "Transition density: A new measure of activity in digital circuits," *IEEE Transactions on Computer-Aided Design*, pp. 310–323, Feb. 1993.

[2] A. Ghosh, S. Devadas, K. Keutzer, and J. White, "Estimation of average switching activity in combinational and sequential circuits," *29th ACM/IEEE Design Automation Conference*, pp. 253–259, June 8–12, 1992.

[3] M. Xakellis and F. Najm, "Statistical Estimation of the Switching Activity in Digital Circuits," *31st ACM/IEEE Design Automation Conference*, pp. 728–733, 1994.

[4] F. Brglez, P. Pownall, and R. Hum, "Accelerated ATPG and fault grading via testability analysis," *IEEE International Symposium on Circuits and Systems*, pp. 695–698, June 1985.