

# A Worst-Case Circuit Delay Verification Technique Considering Power Grid Voltage Variations

Dionysios Kouroussis, Rubil Ahmadi and Farid N. Najm  
Department of ECE, University of Toronto, Ontario, Canada  
(diony,rubil)@eecg.utoronto.ca, f.najm@utoronto.ca

*Abstract*—

In the verification of VLSI circuit design, Static Timing Analysis (STA) techniques allow a designer to calculate the timing of a circuit at different process corners which only consider cases where all the supplies are low or high. This analysis may not be the true maximum delay of a circuit due to the neglect of mismatch between drivers and load. We propose a new methodology for timing analysis where we identify all the possible critical paths of a circuit using new timing models while integrating the aforementioned mismatch for the logic gates. Given then these critical paths we tie the supplies of the gates to physical power grids and re-analyze for the worst-case time delay. This re-analysis is posed as a sequence of optimization problems where the complete operation of the entire circuit is abstracted in terms of current constraints. We present our technique and report on the implementation results using benchmark circuits tied to a number of test-case power grids.

## I. Introduction

In the analysis and verification of high performance chip-design it is essential that timing analyzers take into account power supply variations. The task however is difficult, due to the increasingly large size of power grids and the difficulty of obtaining all possible circuit behaviors especially those that will produce an accurate worst-case time delay of a circuit.

Voltage fluctuations within on-chip power grids, are a result of many sources such as  $IR$ -drop,  $Ldi/dt$  drop, and resonance between the grid and the package. Typically at frequencies below a GHz or so, the inductance is neglected and simulation of the power grid is focused on only  $IR$ -drop given the  $RC$  structure of the grid. To simulate the voltage drop on the grid, designers provide some form of current profiles of the behavior of the circuit. These profiles are then used to calculate voltage drop on grid. Due to the very large number of circuit behaviors, however, it is impractical to simulate the circuit (for the currents) and the grid (for the voltage drops) for all possible clock cycles or vector sequences.

We are interested in the possibility of determining how the voltage fluctuations of the grid will effect timing delay without having complete knowledge of the circuit behavior. We will assume that only *incomplete* information of circuit behavior is available in terms of circuit currents. This incomplete circuit current information will be in the form of current constraints as

presented in [2]. These constraints essentially are the upper bounds on the currents.

Further, we study the effect of variations of the grid voltages on the circuit timing, and develop a static timing analysis (STA) approach that takes these variations into account. We begin by assuming that the exact voltage drops are not known, but that the *ranges* of voltage drops are specified. As well, we assume that the voltage drops are independent in order to identify the *worst-case* voltage configuration that causes a logic circuit to exhibit its worst-case delay. Once we have obtained the worst-case configurations and compared critical paths we tie the supplies of the gates in the worst paths to a test-case power grid and analyze the time delay based on the actual abstracted working behavior of the circuit as represented by the current constraints. The delay time re-analysis is done by using the delay model developed in [1]. This model was formulated into a nonlinear programming problem, which we solved for the maximum delay subject to the current constraints.

## II. Methodology Overview

Here we will quickly outline our proposed timing verification technique:

1. abstract the entire behavior of circuit in terms of current constraints.
2. extract and prioritize the critical paths of the circuit using upper/lower bound supply variations
3. verify voltage of power grid supply nodes of the critical paths to be within bounds.
4. solve for the maximum time delay of paths given current constraints.

## III. Power Grid Modeling

In order to calculate the voltage drops on grid, we will use the revised system of Modified Nodal Analysis (MNA) [4] :

$$\mathbf{G}\mathbf{v}(t) + \mathbf{C}\dot{\mathbf{v}}(t) = \mathbf{i}(t) \quad (1)$$

where  $\mathbf{G}$  is an  $n \times n$  conductance matrix,  $\mathbf{C}$  is an  $n \times n$  diagonal matrix of node capacitances, and  $\mathbf{v}(t)$  the vector of voltage drops at the associated nodes, while  $\mathbf{i}(t)$  is the vector of currents being supplying the circuit.

In (1) one can solve directly for the voltage drop values. The circuit described by these equations consists of the original power grid, but with all the voltage sources set to zero (short-circuit) and all the current source directions reversed.

Based on the monotonicity property of the power grid, [5] [3] [2], we can make a couple of statements that are useful. Let  $I_k$  be an upper bound on  $i_k(t)$  over the time period of interest, say  $0 \leq t \leq \infty$ . Let  $I_1, I_2, \dots, I_n$  form a  $n \times 1$  vector  $\mathbf{I}$  and let  $\mathbf{V}$  be the solution of the system when the DC currents  $\mathbf{I}$  are applied as inputs, which may be found by solving the DC system:

$$\mathbf{G}\mathbf{V} = \mathbf{I} \quad (2)$$

Then, from the monotonicity property, it is clear that  $\mathbf{i}(t) \leq \mathbf{I}, \forall t \geq 0$  leads to  $\mathbf{v}(t) \leq \mathbf{V}, \forall t \geq 0$ . Finally, another related result is that, considering the DC system (2), if  $\mathbf{I}^* \geq \mathbf{I}$ , then  $\mathbf{V}^* \geq \mathbf{V}$ .

#### IV. Peak Current Constraints

In order to abstract the behavior of the entire chip we will use the two related notions of an incomplete current specification, referred to as *current constraints: local constraints*, and *global constraints* as formulated in [2].

##### A. Local Constraints

A local constraint relates to a single current source. For instance, one may specify that current  $i_k(t)$  never exceeds a certain fixed level  $I_{L,k}$ , i.e.,  $i_k(t) \leq I_{L,k}, \forall t \geq 0$ . This upper bound may be simply known from prior simulation, if the cell or block is already available, or it may be a best-guess based on the area of the cell or block and on perhaps the *power density* of the design (total power divided by total area). We express these constraints in vector form as:

$$\mathbf{0} \leq \mathbf{i}(t) \leq \mathbf{I}_L, \forall t \geq 0 \quad \text{or} \quad \mathbf{0} \leq \mathbf{i}(t) \leq \mathbf{i}_L(t), \forall t \geq 0 \quad (3)$$

With only a notion of local constraints however, the voltage drop on grid would be very pessimistic, since it is never the case that all chip components draw their maximum current simultaneously. Thus, it is necessary to include some form of global constraints.

##### B. Global Constraints

It is also useful to express constraints related to all current sources or to sub-groups of current sources. For instance, if the total power dissipation of the chip is known, even approximately, then one may say that the sum of all the current sources is no more than a certain upper bound. We refer to this type of constraint as a *global constraint*. In general, a global constraint corresponds to the case when the sum of the currents for a group of current sources is specified to have an upper bound. If  $m$  is the number of available global

constraints, then we express all the global constraints in matrix form as:

$$\mathbf{U}\mathbf{i}(t) \leq \mathbf{I}_G \quad \text{or} \quad \mathbf{U}\mathbf{i}(t) \leq \mathbf{i}_G(t) \quad (4)$$

where  $\mathbf{U}$  is a  $m \times n$  matrix that contains only 0s and 1s.

##### C. Combining Constraints

The local and global constraints can be combined into a single matrix inequality, as follows:

$$\begin{aligned} \mathbf{L}\mathbf{i}(t) \leq \mathbf{I}_m \quad \text{or} \quad \mathbf{L}\mathbf{i}(t) \leq \mathbf{i}_m(t), \\ \text{with } \mathbf{i}(t) \geq \mathbf{0}, \quad \forall t \geq 0 \end{aligned} \quad (5)$$

where  $\mathbf{L}$  is an  $(n+m) \times n$  matrix of 0s and 1s, whose first  $n$  rows form an identity matrix (1s on the diagonal and 0s everywhere else) and whose remaining  $m$  rows correspond to the matrix  $\mathbf{U}$ , and where  $\mathbf{I}_m$  and  $\mathbf{i}_m(t)$  are  $(n+m) \times 1$  vectors.

##### D. Voltage Formulation

By making use of the relationship  $\mathbf{I} = \mathbf{G}\mathbf{V}$ , we can express the DC constraints in terms of voltages:

$$\mathbf{L}\mathbf{G}\mathbf{V} \leq \mathbf{I}_m, \quad \mathbf{V} \geq \mathbf{0} \quad (6)$$

#### V. Time Delay Modeling

We use the time delay modeling developed in [1]. In this way we have an accurate time delay for logic cells that is dependent on power supply and ground voltage fluctuations.

Modern cell libraries represent the delay of cells using four 2-dimensional tables for each timing arc (a timing arc is an input-output node pair). In case of a falling output, one table gives the propagation delay and another gives the output slope. Another two tables correspond to the rising output case. Each table covers the range of valid input slope and output load values. Simple extension of this model to the model in [1] would require 6-dimensional tables, which would be impractical in terms of model size and cost of building the model. In order to simplify the model, it was found the delay dependence on each voltage is near-linear in the (narrow) range of valid voltages. However, to be more accurate, a quadratic polynomial representing the dependence of delay on each voltage was used, and allowance was made for cross-product terms, by using a template expression for delay as follows:

$$\begin{aligned} t_d = \sum_k \alpha_k V_{ih}^{a_k} V_{il}^{b_k} V_{dd}^{c_k} V_{ss}^{d_k} \\ \text{where } \alpha_k \in \mathcal{R}, \quad \text{and} \\ a_k, b_k, c_k, d_k \in \{0, 1, 2\} \end{aligned} \quad (7)$$

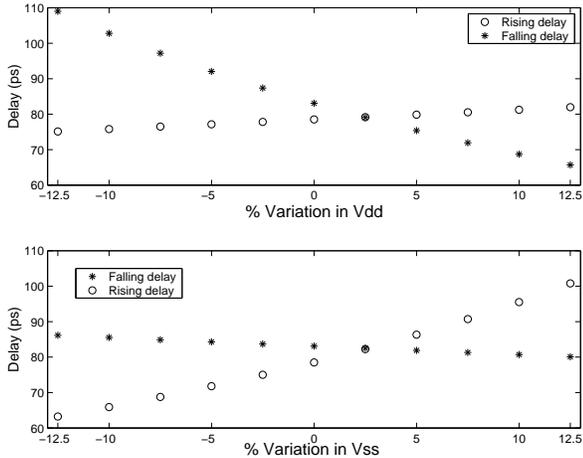


Fig. 1. Delay vs. Voltage of a two gate system.

The regression coefficients  $\alpha_k$  were found by using a standard Least Mean Square (LMS) regression method [6]. The regression is performed for each grid point in the [slope, load] table, so that each cell in the [slope, load] table contains the values for a number of coefficients  $\alpha_1, \alpha_2, \dots, \alpha_m$ .

## VI. Worst-Case Delay with Grid

Having found an accurate model of path delay in terms of supply voltage fluctuations, we are now interested in finding the maximum possible delay, given the DC current constraints which abstract the full operation of the entire circuit. We may pose the worst case time delay as a nonlinear programming (NLP) problem in the form of:

$$\begin{aligned} & \text{maximize :} && t_d && (8) \\ & \text{subject to :} && \mathbf{LGV} \leq \mathbf{I} \\ & && \mathbf{V} \geq \mathbf{0} \end{aligned}$$

Here it becomes of interest to verify or prove that the solution of  $t_d$  is in fact a global optimum as opposed to some local optimum that is found given the linear constraints. It is hard to analytically prove that the solution is a global optimum but we may justify it in the following manner.

Firstly the current constraints of our problem have to be verified in such a way that they do not produce voltages on the grid that exceed the bounds of the worst case voltage used in the development of the time delay model. This may be easily done using an interior point method maximizing the voltage drop of the connecting supply nodes of the critical path. Once the supply nodes have been verified, that they are within the bounds of the model, then we may proceed to our nonlinear optimization.

So now we are certain that the bounds of the voltages are within our model parameters. How can we show

that the optimization will produce a global solution for the time delay? We need to look at the sensitivity analysis of the time delay model. We know that for any path, a supply voltage will only effect 2 gates (*or blocks*) in the path namely the gate it is directly connected to, and the gate that is driven by the supply connected gate.

Here, we check the delay of two connected inverter gates with independent supplies and grounds in presence of 12.5% voltage variation around the nominal values. Figure 1 shows that the delay of two consecutive gates is always monotone due to variation of power supply and ground voltages and the sensitivity is positive or negative respective to the signal polarity. Delay sensitivities of all gates and gate combinations in our library have been checked and it is confirmed that the sensitivity of the delay to a given voltage variable does not change sign as that voltage is varied across the whole range.

Given the fact that empirical data shows that delay has a monotonous curve versus each of the voltages and that the maximum delay occurs at the corner of the voltage domains we may safely state that optimizing the above expression under the linear constraints, which produce voltage bounds that are within model parameters, will provide a local solution that is also the global solution.

### A. Supply and ground node verification

We need to verify that the voltages of the supplies feeding the path are within bounds. This is done by firstly extracting the set of nodes that supply the path ( $v_i, v_j, \dots, v_m$ ). Then we may formulate the sequential linear program:

$$\begin{aligned} & \text{maximize :} && v_i, v_j, \dots, v_m && (9) \\ & \text{subject to :} && \mathbf{LGV} \leq \mathbf{I} \\ & && \mathbf{V} \geq \mathbf{0} \end{aligned}$$

If the solution set of maximized voltages is within the model bounds we may proceed with our nonlinear optimization (maximization) of the time delay. Here we need to keep in mind that the solution of worst-case voltage at each node is the absolute worst case under all possible circuit conditions. If the voltages exceed the model bounds then the grid needs to be corrected and the supply nodes verified again.

### B. Time Delay Analysis

As we have shown that our NLP will produce a local optimum that is also the global optimum because of the monotonicity of the time delay function, our NLP solver only needs to find a local solution. We use the SNOPT solver for our optimizations which can only find local solutions. We generate the functions of all

TABLE I  
METHODOLOGY RESULTS.

Circuit	# of paths extracted	average # of gates/path	worst-case $t_{d(STA)}$ (ns)	# of Powergrid nodes	worst-case $t_{d(NLP)}$ (ns)	# of paths analyzed	Difference STA vs. NLP	NLP solution time (sec.)
C1345	1	28	4.54	1320	4.13	1	9.9%	0.61
C1908	7	43	5.90	1320	5.31	6	11.1%	4.02
C2607	3	36	5.82	1320	5.64	1	3.2%	0.64
C3540	4	48	7.43	5832	6.96	1	6.7%	1.37
C432	5	43	7.22	5832	7.10	1	1.7%	1.54
C499	1	26	3.64	10073	3.52	1	2.6%	1.69
C5315	5	42	6.79	10073	6.28	3	8.1%	5.13
C7552	5	37	5.48	27055	4.84	3	13.2%	8.80
C880	7	25	3.92	27055	3.62	3	8.3%	8.34
S420	5	11	2.13	27055	1.81	2	11.9%	5.12
S510	4	11	2.14	43106	1.77	1	20.9%	3.87
C6288	14	113	21.47	43106	18.80	9	14.2%	32.10

the gradients of our objective to improve its efficiency. Since only our objective is nonlinear, the problem is linearly constrained which tends to solve more easily than general nonlinear programs with nonlinear constraints.

## VII. Experimental results

Our technique was implemented and tested on the IS-CAS85 and the combinational parts of the ISCAS89 benchmarks, using randomly generated power grids. Not having access to power grids from industrial designs and in order to test our approach under different conditions, we have opted to *generate* a number of grids ourselves. The supplies of the critical paths extracted from the ISCA benchmarks were then randomly connected to our power grids. This random process of circuit to power grid connection was done in order to best emulate all the possible designs that could be encountered from critical paths within specific blocks to paths that may span the geometry of the entire chip.

Experiments were run on a 1 GHz Sun machine with 4 GB memory. Table I shows some of our results. A number of benchmark critical paths randomly connected to varying sized power grids, from 1000 nodes to 40,000 nodes, were simulated using our NLP approach. The worst case delay found under the influence of power grid is smaller than that found using STA analysis with independent supplies. The difference is seen to vary between 2% to 20%. The computation time for the NLP solves of the critical paths analyzed, are a minimal 1 to 30 seconds. This reported time is only the time required to solve for the maximum (NLP) time of the critical paths. It does not include the time required to perform a *preconditioning* on the linear component of the problem which may run in the order of 10 to 15 minutes for the larger sized grids. Further, it was observed that our technique used about 100Mb of memory for the large grids, thus, it may be easily applied to even larger grids. We do not formally report on the computational cost times of the node voltage bound verification, except to say that the method in [2] has

been improved upon significantly by implementing an Interior Point Method with sparse storage techniques. The time required for one check of a node voltage is in the order of half a minute for a 40,000 node grid. This check may be easily expanded to larger grids as well.

## VIII. Conclusion

In todays integrated designs, timing and its sensitivity to supply voltage fluctuations is a major concern for design closure. We have proposed a method where by we abstract circuit behavior in the form of user-supplied current constraints. By using a delay model that is expressed in the form of supply voltage variations of the path and running a nonlinear program, we may solve for the worst-case time delay. This delay is a more precise measure than what the current STA tools can provide using only nominal or worst-case voltage levels.

## References

- [1] R. Ahmadi and F. N. Najm. Timing analysis in presence of power supply and ground voltage variations. In *International Conference on Computer Aided Design*, San Jose, 2003.
- [2] D. Kouroussis and F. N. Najm. A static pattern-independent approach for power grid voltage integrity verification. In *40<sup>th</sup> Design Automation Conference*, Anaheim, 2003.
- [3] H. Kriplani, F. N. Najm, and I. Hajj. Pattern independent maximum current estimation in power and ground buses of CMOS VLSI circuits: algorithms, signal correlations, and their resolution. *IEEE Transactions on Computer-Aided Design*, 14(8):998–1012, August 1995.
- [4] L. T. Pillage, R. A. Rohrer, and C. Visweswaraiiah. *Electronic Circuit and System Simulation Methods*. McGraw-Hill, Inc., New York, NY, 1995.
- [5] J. Rubenstein, P. Penfield, and M. A. Horowitz. Signal delay in RC tree networks. *IEEE Trans. on Computer-Aided Design*, 2(3):202–211, July 1983.
- [6] B. Widrow. *Adaptive signal processing*. Prentice-Hall, 1st edition, 1985.