INTEGRATION, the VLSI journal 45 (2012) 365–375

Contents lists available at ScienceDirect

INTEGRATION, the VLSI journal

journal homepage: www.elsevier.com/locate/vlsi

An accurate sparse-matrix based framework for statistical static timing analysis[☆]

Anand Ramalingam[a], Ashish Kumar Singh[d,*], Sani R. Nassif[c], Gi-Joon Nam[c], Michael Orshansky[b], David Z. Pan[b]

[a] *Magma Design Automation, Austin, TX 78759, United States*
[b] *Department of Electrical and Computer Engineering, The University of Texas, Austin, TX 78712, United States*
[c] *Austin Research Lab, IBM, Austin, TX 78758, United States*
[d] *Terra Technology, Schaumburg, IL, 60173, United States*

ARTICLE INFO

Article history:
Received 1 July 2010
Received in revised form
17 November 2010
Accepted 1 March 2011
Available online 21 March 2011

Keywords:
Statistical timing
Sparse matrix
Path-based timing

ABSTRACT

Statistical static timing analysis has received wide attention recently and emerged as a viable technique for manufacturability analysis. To be useful, however, it is important that the error introduced in SSTA be significantly smaller than the manufacturing variations being modeled. Achieving such accuracy requires careful attention to the delay models and to the algorithms applied. In this paper, we propose a new sparse-matrix based framework for accurate path-based SSTA, motivated by the observation that the number of timing paths in practice is sub-quadratic based on a study of industrial circuits and the ISCAS89 benchmarks. Our sparse-matrix based formulation has the following advantages: (a) it places no restrictions on process parameter distributions; (b) it can use an accurate polynomial-based delay model which takes into account slope propagation naturally; (c) it takes advantage of the matrix sparsity and high performance linear algebra for efficient implementation. Our experimental results are very promising.

1. Introduction

As technology has scaled, manufacturing variations have emerged as a major limiter of design performance. These variations exhibit themselves as systematic, spatial and random changes in the parameters of active (transistor) and passive (interconnect) components. Furthermore, these variations are increasing with each new generation of technology. Statistical static timing analysis (SSTA) has been proposed to perform full-chip analysis of timing under such types of uncertainty, and has been the subject of intense research recently [2–19]. The result of SSTA is the prediction of parametric yield at a given target performance for a design.

SSTA algorithms can be classified into two major groups:

(1) *Block-based* [2–7] approaches use a breadth-first traversal of the circuit to compute circuit delay [2]. The delay pdf is propagated from the primary inputs to the primary outputs. The major difficulty in block-based approaches is the introduction of the max operator at each block, and the need to accurately estimate the maximum of two random variables in the same form in which those two variables are defined.
(2) *Path-based* [8–12] approaches rely on an enumeration of all or a large number of the most critical paths in the circuit [8]. Considering the case where all paths are enumerated, the max operator is deferred to the end of the analysis (i.e. taking the maximum of all the paths) and therefore does not introduce any inaccuracy in the computation. A major problem with path-based approaches is the *perception* that typical circuits have an exponential number of paths, making the computational requirement for such approaches impractical.

While there has been much work on the algorithms for SSTA, there has been somewhat less work on the accuracy issues. Some of the sources of inaccuracy in SSTA are

- the basic assumptions underlying static timing analysis, such as treating a gate as a node without considering the functionality which gives rise to false paths,
- the delay models used for gates and wires, and
- the model for process variations and their spatial and/or temporal distributions.

The *algorithmic* error introduced by SSTA algorithms can be traced back to the application of the max operator, which is an

[☆] A preliminary version of this paper appeared in the Proceedings of the International Conference on Computer-Aided Design (ICCAD 2006) [1].

[*] Corresponding author. Tel.: +1 5126892111.

E-mail addresses: anandr@magma-da.com (A. Ramalingam), aksingh@cerc.utexas.edu (A.K. Singh), nassif@us.ibm.com (S.R. Nassif), gnam@us.ibm.com (G.-J. Nam), orshansky@cerc.utexas.edu (M. Orshansky), dpan@cerc.utexas.edu (D.Z. Pan).

0167-9260/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.vlsi.2011.03.002

approximation to the behavior of true circuits, and which is further approximated in SSTA algorithms [14–16]. While a direct assessment of that error is difficult, we propose that eliminating the max operation on parameterized form would aid in reducing the error. The algorithm we propose in this work applies max on a set of real numbers thus incurring no error.

The *model* error has been widely recognized and a number of researchers have made important contributions. The *parameterized delay form* expressed delays and arrival times as explicit linear functions of the process parameters [6]. It was later expanded to handle quadratic delay models that are able to improve the accuracy of delay estimates [13–16]. A related source of error, namely the modeling and handling of the slope of signals, has not received as much attention. In fact, current published approaches typically make a worst-case estimate of the slope or propagate the latest arriving slope [6] which can lead to significant error [20]. The polynomial models we propose in this work allow high accuracy by using higher order models, and naturally handling the slope and its propagation.

The *distribution* error, i.e. the error caused by lack of generality in the modeling of the statistical properties of the process variations has been the most difficult to deal with, due to the lack of published realistic manufacturing variability data. Earlier approaches assumed that process variables followed normal distributions [8], but recent work has shown how more general distributions can be handled, and how spatial and systematic correlation can be accommodated [19]. In this work, we make *no assumptions* about the character or distribution of any process parameter.

This paper proposes a new approach to parameterized path-based SSTA. The proposed method starts with a preprocessing step of path enumeration and delay computation of all the paths in a parameterized form, which we then efficiently represent using a *sparse matrix*. We model the delay and slope of each component in the circuit using a general parameterized polynomial form which can include the influence of

- input waveforms and output loading,
- manufacturing variations in parameters like threshold voltage and channel length,
- operating environment variations in parameters like power supply voltage and temperature.

Next, the path delays in this same parameterized form are computed by a natural extension to the gate delay formulation. Given a sample of values from the distribution of manufacturing variations, this computation is shown to be simply a matrix/vector multiply that produces a vector of delays of each path in the circuit. Finally, the maximum circuit delay is obtained by applying the max operator on the path delays. The major attributes of this work are

(1) We show that the number of paths in practice is sub-quadratic in number of gates by evaluating the number of paths in the ISCAS89 benchmarks as well as two different families of industrial circuits.
(2) It can handle global, spatial and intra-die variations in one unified framework.
(3) It can compute the delay based on an accurate propagation of slope along all paths.
(4) It minimizes the impact of the error caused by approximating the max function commonly used in SSTA.
(5) It is independent of the underlying distribution of the process parameters, and is not restricted to the usual Gaussian distribution.

The remainder of this paper is organized as follows. In Section 2 we first motivate the path-based approach by showing that it is indeed practical for many circuits. We then show the need for higher order (more than linear) delay models in Section 3, and describe our approach to the delay modeling of practical static CMOS gates. With those models in hand, we then describe our matrix based formulation for STA and SSTA without slope in Section 4 and SSTA with slope in Section 5. We demonstrate its application to the ISCAS family of sequential benchmark circuits in Section 6. We compare our method with an existing method in Section 7 and conclude in Section 8.

2. A case for path-based SSTA

The upper bound on the number of paths in an arbitrary network is exponential in the number of gates. A key observation in this paper, however, is that for the vast majority of practical circuits, the number of actual paths is far less than this theoretical upper bound, and is quite manageable. It should be noted that the theoretical upper bound is usually met by highly structured networks such as multipliers. With the easy availability of large amounts of memory in modern computers, storing and manipulating million of paths is eminently practical.

To test our conjecture, we enumerated *all* the latch-to-latch, primary input to latch, and latch to primary output paths in the ISCAS sequential circuit benchmarks [21] (see Fig. 1), and found that the paths $\approx 0.04 \times \text{gates}^{1.8}$. This is hardly the type of explosive growth that might cause one to completely discount a family of algorithms. But since the ISCAS benchmarks are small compared to modern designs, we further extended our analysis to two different families of industrial benchmarks, one for large circuits (much larger than the ISCAS benchmarks), and one for moderate sized circuits (comparable to the ISCAS benchmarks).

We enumerated all paths for the circuits in those two families. For the first and larger family, shown in Fig. 2, we saw that the number of paths $\approx 0.12 \times \text{gates}^{1.42}$. For the second and smaller family, shown in Fig. 3, we found paths $\approx 0.43 \times \text{gates}^{1.17}$.

Clearly, the demonstration above should not be taken as sufficient license to propose a purely path-based SSTA algorithm. However, it does demonstrate that such an algorithm can be practical for a significant number of cases. In the broader picture, one can imagine a pairing of path-based and block-based algorithms with one being applied when the enumeration of paths results in a manageable number of paths, while the other gets applied to those circuits where the number of paths exceeds some suitable threshold.

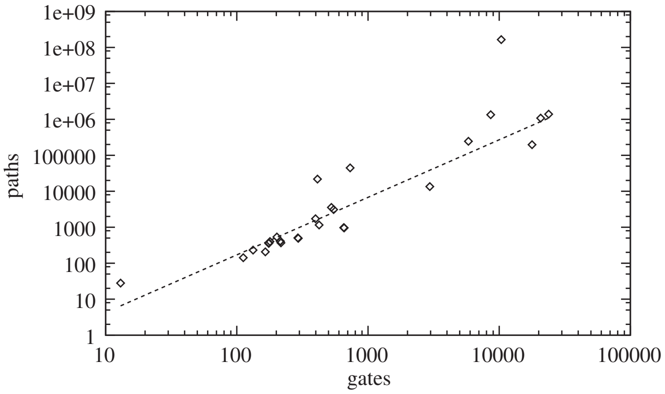

Fig. 1. The number of paths versus the number of gates in ISCAS'89 benchmarks. By linear regression we get the following relationship: paths $\approx 0.04 \times \text{gates}^{1.8}$.