Capturing variability in advanced gate delay models[1]

Prasanjeet Das and Sandeep K. Gupta

Department of Electrical Engineering – Systems

University of Southern California

Los Angeles CA 90089

prasanjd@usc.edu, sandeep@usc.edu

Abstract— All post-silicon tasks – validation, diagnosis, delay testing and speed-binning – must be carried out by applying vectors to actual chips, and capturing and analyzing responses. Yet, vectors used must be generated and analyzed using pre-silicon models of the circuit. Three comprehensive industrial studies demonstrate that existing approaches for generating and analyzing such vectors are inadequate, and one major weakness is that existing delay models *either* do not capture process variations *or* do not capture advanced delay phenomenon that significantly affect delays, especially multiple input switching (also known as near-simultaneous transitions) at inputs of a gate, the charge on internal capacitance and Miller effect. In this paper, we address two related questions. (1) How do we extend advanced delay models that capture these delay phenomena to also capture process variations? (2) How do we use such a model, especially to select paths and generate or evaluate vectors for post silicon tasks? In particular, we present a general approach for extending any delay model (pin-to-pin and beyond) to ensure that all minimum and maximum delay values computed are guaranteed to bound the corresponding delay values in silicon. We present extensive experimental results to demonstrate that our models capture variability in a way that tightly bounds the actual delays, at a computational complexity that is practical even for generating vectors.

Keywords: multiple input switching, process variations, delay models, delay marginality, first-silicon validation and speed binning.

[1] This research was supported by Intel Corporation.

I. INTRODUCTION

Timing simulation and analysis has become central to any design flow - starting from design specifics to high volume manufacturing (see Table I). On pre-silicon front, circuit timing simulation (with fully specified vectors) and static timing analysis (with fully unspecified vectors), have become imperative for design tasks such as retiming and synthesis. Whereas, on post-silicon front, the vectors (that are generated based on the pre-silicon timing analysis - which is typically invoked a large number of times during vector generation with fully specified, partially specified and unspecified vectors) are applied for tasks such as delay validation, diagnosis, testing and speed-binning. It is usually desired, that the post-silicon delay estimates must strongly match the pre-silicon ones, but unfortunately, this seldom happens in industrial practice [1][2][3].

Delay testing is the basic technique used for identifying slow paths and slow ICs. In slightly different variations it is used during validation, diagnosis and characterization of the first-silicon for a new design. It is also used after the design moves into high volume manufacturing, during delay testing and speed binning [3].

One important common characteristic of all the above post-silicon tasks is that they are vector based and require us to generate vectors that will provoke worst-case delays, evaluate given vectors in terms of their ability to excite high delays, analyze vectors that fail tests at high speeds to identify the root causes – namely slow sub-paths or gates, and so on.

TABLE I: Tasks that require timing

Tasks	**Pre-silicon/Post-silicon**	**Vector based**		
		Fully-specified	**Partially-specified**	**Unspecified**
Circuit simulation	Pre-silicon	Y		
Static Timing Analysis	Pre-silicon			Y
Retiming and Synthesis	Pre-silicon	Y		Y
Vector generation (validation, diagnosis, testing, speed binning)	Pre-silicon	Y	Y	Y
Vector application (validation, diagnosis, testing, speed binning)	Post-silicon	Y		

Several recent industrial case studies on characterization of timing behavior using fabricated chips from nVidia, Freescale, and Sun (now Oracle) [1][2][3] show that *existing path delay testing (PDT) approaches generate vectors that fail to invoke the worst-case delays in silicon*. Also, one common observation from [1][2][3] is that the *PDTs invoke lower delays than functional tests since the critical paths in silicon are different from those identified by existing static timing analysis (STA) approaches*.

These results are contrary to common belief that path delay testing approaches do capture worst-case delays. Hence, above three studies [1][2][3] have further investigated this and concluded that this contradiction is mainly due to the limitations of the delay models.

Pre-silicon delay models are foundations of all variants of delay testing for the above post-silicon tasks, since these tasks use pre-silicon models to select/prioritize paths based on their delays, generate suitable vectors, and analyze given vectors. As the fabrication process moves into nano-scale, the importance of many delay phenomena [4][5] and levels of process variations [6]-[9] are growing. These two facts are making delay models from recent past increasingly *inaccurate* (i.e., unable to capture emerging delay phenomena) and *non-resilient* (i.e., invalidated by process variations). Delay model **inaccuracy and non-resilience** are the two main reasons behind the limitations of the above silicon studies [1][2][3]. Moreover, an inaccurate and non-resilient delay model can diminish validation quality; also it can increase the number of vectors generated and hence increase validation costs. Similarly, such a delay model can decrease the resolution of delay *diagnosis* and hence increase the costs of redesign, and reduce the confidence in *speed binning*.

All the above post-silicon tasks require delay models that are accurate and resilient. Any vector generation approach – for validation, delay testing and speed-binning – starts with a completely unspecified vector and successfully specifies additional bits of the vector. Since only fully-specified vectors can be applied during post-silicon tasks, all these tasks are carried out using fully-specified vectors. Hence, *post-silicon tasks require a delay model that can work with fully- and partially-specified as well as fully-unspecified vectors.* Finally, since the delay engine is called frequently during vector generation and simulation in a timing-oriented framework, *all post-silicon tasks require delay models that have low computational complexities*.

Although many approaches [10]-[26] have improved accuracy and resiliency of delay models used for pre-silicon timing analysis, *delay models that satisfy all the above requirements of post-silicon tasks do not exist in practice*. (More details in Section II.) The objective of this paper is to develop a delay model that is resilient and, at low computational complexity, provides accuracy for partially- and fully-specified as well as fully-unspecified vectors. We achieve this combination of goals by starting with ***an existing accurate delay model*** [4][5] which captures most of the known delay phenomena in modern CMOS technologies and using ***bounding approximations*** that use piecewise-linear bounding functions to capture process variations and inaccuracies. Our accurate and resilient delay model is compatible with some existing approaches for timing analysis [27][28]. Elsewhere we show how our delay model can be used to generate vectors for post silicon delay marginality validation [29].

This paper is organized as follows. In Section II, we present an extensive survey of the research literature and present the motivation for and the importance of our approach. In Section III, we outline our overall approach. The experimental setup and results are described in Section IV. Finally, conclusions are presented in Section V.

II. MOTIVATION

Delay models of gates and wires are used by static timing analysis (STA) to estimate the performance of a chip before releasing its design for fabrication (tapeout). The correlation of the STA results with measurements on actual silicon is primarily determined by accuracy of delay calculations [9]. It is imperative for a gate's delay model to accurately represent the logic as well as timing behavior of the gate. A *basic delay model* considers basic delay determinants of the gate, such as input slew and output load. These models are extended to derive *advanced delay models* that capture additional phenomena associated with timing behavior. The timing behavior are often classified into three – single input switching (SIS), multiple inputs switching for to-controlling (MIS-TC) transitions, and multiple inputs switching for to-non-controlling (MIS-TNC) transitions. It is now increasingly common for delay models to also account for additional effects, such as crosstalk and ground bounce, and variability [9].

MIS-TC transitions at the inputs of a primitive gate decrease the gate's delay due to activation of multiple charge/discharge paths [4]. On the other hand MIS-TNC transitions at inputs increase the gate's delay due to Miller effect [5]; in this case, the gate's delay also depends on the initial state of the capacitiances of internal nodes between series transistors, body effect, and impedence matching (history or stack effects) [5][16]. Figure 1 shows the *delay vs skew* for near-simultaneous transitions at the inputs of a 2-input NAND gate in 65nm CMOS technology, to–controlling (MIS-TC) and to-non-controlling (MIS-TNC).

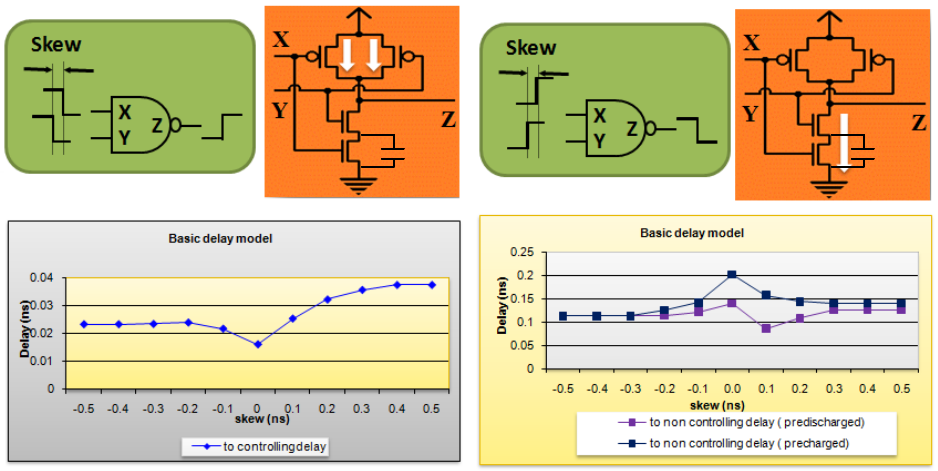


Figure 1: Delay vs Skew curves for MIS [4][5]

In current practice, delay models are categorized into two types - voltage reference models (VRM) and current source models (CSM) (see Figure 2). VRMs define the characteristics of the voltage response at the gate output as a function of input slew and output load using look up tables and interpolation [30], whereas CSMs use a non-linear voltage-controlled current source to determine the output delay and slew via circuit simulation [16].

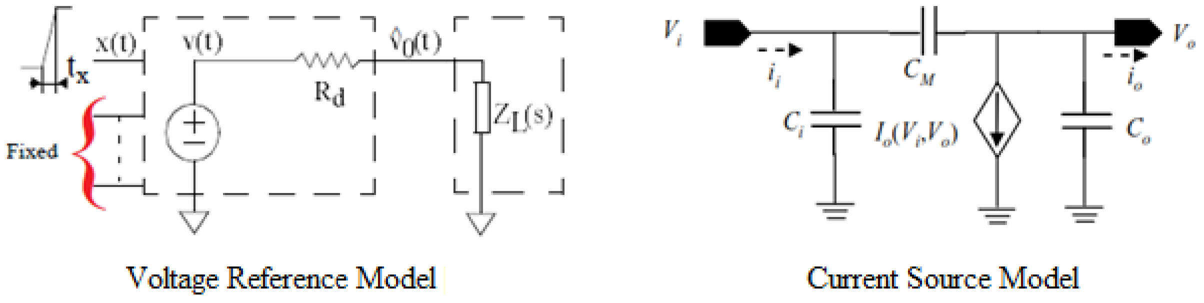


Figure 2: Delay model categorization [30][16]

MIS-TC is a well-researched phenomena and has been considered by many delay models and timing analysis approaches ([13]-[15] for VRMs, and [9][17] for CSMs). MIS-TNC is particularly important since the associated effects (Miller effect, body effect, and stack effects) can ***increase*** gate delays. The models in [4][5] (for VRM) and [16] (for CSM) are the first ones to employ a model that combined MIS-TC and MIS-TNC (see Table II).

Although industry has migrated from VRMs (SPDM (Scalable Polynomial Delay Model (Synopsys)) to CSMs such as ECSM (Cadence) and CCS (Synopsys) for pre-silicon design oriented tasks namely retiming and synthesis because of the latter’s ability to handle complex waveform shapes [9][12], VRMs should be preferred over CSMs for post-silicon tasks due to the associated lower characterization efforts and lower computational complexities (see Section III.F), and advanced delay models, such as those presented in [4][5], are imperative for accurate timing tools [27]. But with the increase in process variations [6]-[8], even these delay models have become inadequate for post silicon tasks for high performance circuits [1][2][3].

Moreover, CSM based models are good for timing analysis only when the input vectors are completely specified, but these models cannot be used for test generation and other tasks which deal with partially specified vectors [27]. To the best of our knowledge, exiting CSM based approaches for Timing Analysis [22][34] are sort of SPICE simulation replacements which cannot work efficiently for the generic static timing analysis [33] to calculate arrival and transition time ranges when the vectors are completely unspecified.

TABLE II: Review of delay models

Reference #	VRM	CSM	SIS	MIS-TC	MIS-TNC	Variation-aware
[4][5]	Y			Y	Y	N
[10][11][12]		Y	Y			N
[13][14][15]	Y			Y	N	N
[16]		Y		Y	Y	N
[17]		Y		Y	N	N
[18]	Y		Y			N
[20][21][22]		Y	Y			Y
[23][24][25][26]	Y			Y	N	Y

The growing effect of variability on delay has fueled the development of statistical delay models [19]. Statistical delay models for CSMs [20]-[22] do not capture MIS effects; those for VRMs consider MIS-TC ([23]-[26]) but ignore MIS-TNC and associated phenomena. Statistical delay models are unsuitable for post-silicon tasks because of their inability to handle correlation efficiently and due to the enormous complexity associated with using realistic distributions (non-Gaussian) in SSTA [19]. Even if correlations are ignored, simplistic assumptions of independent normal Gaussian distributions are made, these delay models cannot capture all associated delay phenomena as well as variability at practical complexity [19]. These delay models report the delay as a distribution rather than in terms of bounds – the format which is easily understandable by existing timing analysis tools [33]. However, bounds can be considered as a special case (truncated) of distributions, but statistical operations of max and sum on these truncated distributions will make the bounds progressively more and more inaccurate [19]. Also, SSTA based on statistical delay models is inefficient for post silicon tasks due to its inability to take advantage of the additional timing related information associated with partially- or fully-specified vectors. Moreover all existing variation aware delay models are analytical in nature and tend to become more complex and less accurate as device dimensions shrink below 65nm [19]. All this necessitates a resilient delay model that can be suitably used for post-silicon delay characterization.

Any vector generation approach for post silicon tasks such as testing or validation starts with a preprocessing step where the given netlist is converted to a format where complex gates are decomposed into primitive gates. Also, in [29] we have motivated why path delay test (PDT) is a suitable candidate for generating vectors for post-silicon validation for gate-dominated paths. Such paths generally consist of more than 15-20 logic gates [1] and frequently occur in large microprocessor blocks [41] that can be either huge data-path blocks (ALU[40]) or control blocks (such as instruction decoder [42])(Also, effects such as crosstalk which are predominant for wire-dominated paths [3] (such as global wires, bus lines) are not considered here). Hence, post-silicon delay model characterization for PDT needs to be performed for only a few primitive gates (AND/OR/NAND/NOR etc) and the associated characterization effort is much less than a comprehensive full library characterization (generally done for pre-silicon delay related tasks) [17][36].*In this paper we propose a resilient delay model that captures the delay phenomena (including MIS-TC and MIS-TNC), captures variations, produces proven upper and lower bounds to be used for worst-case analysis, and tightens these bounds when logic values at any subset of circuit lines are specified.*

III. PROPOSED APPROACH

Every component of our framework – delay models, timing analysis, and target selection– *is designed to guarantee that we do not miss the worst-case, despite all model, parameter, and variation uncertainties*.

Also, we would like to restate the fact that the timing ranges calculated by resilient delay model are tight except for the inherent sources of looseness - *partially specified vectors, variability, and looseness associated with approximations used to reduce complexity*.

A. Our overall approach

An inaccurate and non-resilient delay model can diminish validation quality, at the same time it can increase the number of vectors generated and hence increase its costs. Hence, we need an accurate and resilient delay model for post-silicon tasks that has the following characteristics:

- Captures known and emerging gate delay phenomena[4][5] as well as variability.
- Only uses bounding approximations to tackle unknowns and any simplifications necessary to make complexity manageable.
- Enables computation of tight timing ranges in an efficient manner (manageable complexity) by allowing a timing analyzer to use all available information about logic values [27] (e.g., a logic value at an internal line that can be proven to be a necessary condition for the task at hand, a given partially-specified vector applied at the inputs, and so on).

B. Characterization setup

Currently the prevalent commercial STA tools employ pin-to-pin delay models [33]. Able to consider input slew and output load, pin-to-pin delay models specify the propagation delay from a given input pin to a given output pin. They implicitly consider the state of internal capacitances since only one such state is possible for each single input transition. But they do not accurately capture the delay for multiple input switching (both MIS-TC and MIS-TNC) and thus are unsuitable for post silicon tasks. The models in [4][5] only uses qualitative information, such as causality and other provable properties of underlying physics such as the effect of near simultaneous transitions (MIS-TC and MIS-TNC) and hence are used as the starting points.

The modeling approach in [4][5] consists of:

- Perform simulations by varying all input parameters (input slews, input skews, and the initial state of internal capacitances) over their typical ranges.
- Quantify the significance of associated delay phenomena for simultaneous transitions from the simulation data.
- Identify appropriate input waveforms that activate each phenomenon.
- Develop an empirical model that identifies input waveforms that excite these phenomena.

The characterization setup shown in Figure 3 can be used to feed the gate with different realistic waveforms with different attributes. The standard delay characterization methodology requires the delay calculation of a gate driving capacitive load to be driven by a load less cell [52]. Thus, the parameters (C_1, S_{in}) of the driver circuit can be changed to generate waveforms with different transition times, skews (between pair of waveforms). The voltage at node N_0 is copied to node N_1 to ensure the load less driver requirement of delay characterization [39]. Note that S_{in} responsible for voltage N_0 is an ideal voltage source, but S_m resulting in N_1 is a controlled voltage source with voltage V (N_1) = V (N_0). Switch SW can be used to get the mirrored (SW=1) or inverted (SW=0) waveform. We also varied the internal capacitances (either precharged or pre-discharged) in our characterization step to account for Stack Effect [5][16]. We varied C_r to account for different capacitive loads on the gate's output.

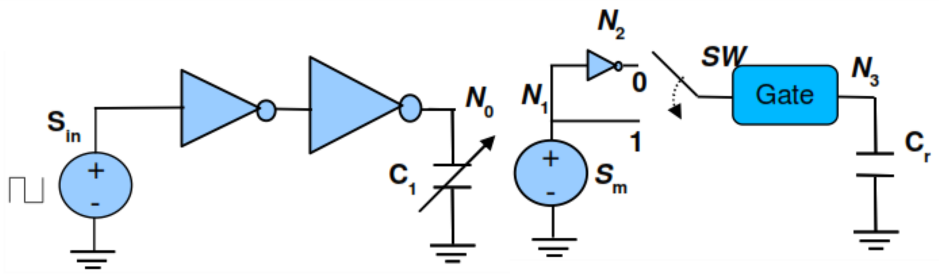


Figure 3: Characterization setup without considering the effect of interconnect.

This characterization setup can be extended to account for interconnect delay using the approach in [36] (see Figure 4). The basic difference between Figure 3 and Figure 4 is the use of a π model load instead of simple capacitive load at driver.

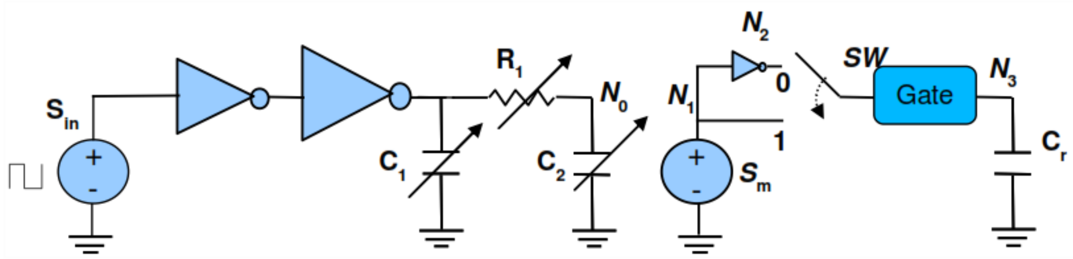
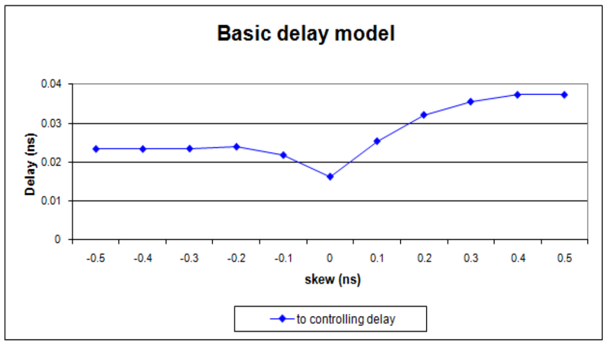


Figure 4: Characterization setup considering the effect of interconnects [36].

C. *Basic delay model*

In order to quantify the significance of associated delay phenomena for the effect of MIS on delay, for a NAND gate in a 65nm CMOS technology we perform circuit-level simulations by varying input transition times (T_R) from 0.1ns to 0.2ns, skew (δ) from -0.5ns to 0.5ns, and output load (C_L) from FO1 to FO4. (We used Spectre for all our simulations.) We arrived at the *delay vs. skew relationship* for near-simultaneous transitions (MIS) at the inputs of a 2-input NAND gate.



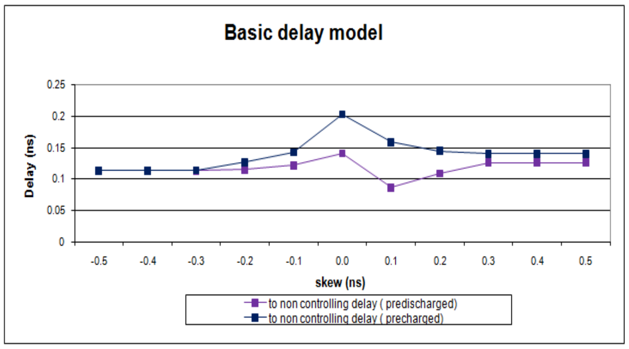
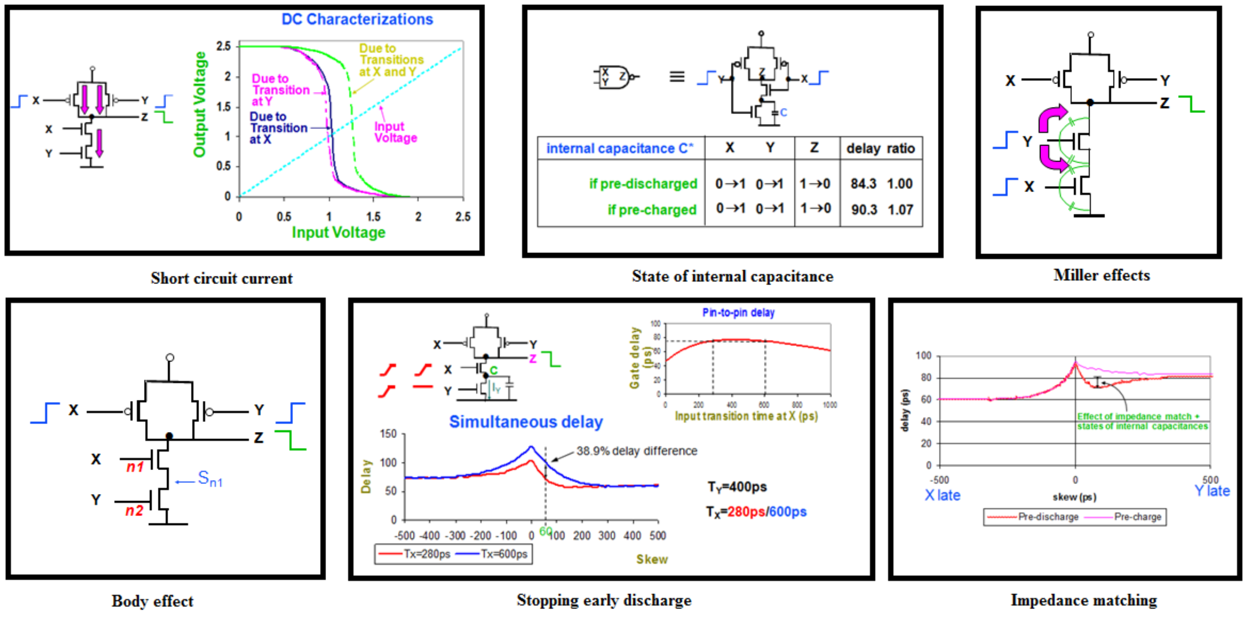


Figure 5: Delay vs skew curve for near-simultaneous transitions (Basic delay model – simulations result)

Figure 5 shows the delay vs. skew relationship (for T^X_R = 100ps, T^Y_R = 100ps, C_L = FO4) for MIS. It is clear from Figure 5 that multiple input switching can significantly affect delay – SIS overestimates MIS-TC by about 30% and underestimates MIS-TNC by about 50% for the cases shown here for 65nm CMOS technology. Similarly, output transition time functions are derived for both MIS-TC and MIS-TNC.

The curves, such as in Figure 5, capture all the known basic delay phenomena associated with MIS. MIS-TC at inputs of a primitive gate decrease gate delay due to activation of multiple charge/discharge paths [4]. On the contrary, MIS-TNC at inputs of primitive gate increase the gate delay [5] due to a combination of various effects such as:

- *Short circuit current* – gate outputs start to switch when the current that charges output is smaller than the current that pulls it down.
- *Initial state of internal capacitances (precharged and pre discharged)* – delay is a function of charge on internal capacitance also know as stack/history effect [16].
- *Miller effect* – Charges are transferred from gate inputs to gate outputs and slow-down output transitions.
- *Body effect* – affects the threshold voltage of a transistor which in turn affects delay.
- *Stopping early discharge* – two skew-delay curves may have same pin-to-pin delay but different simultaneous (MIS related) delay.
- *Impedance matching* – Better match between pull-down transistors can reduce the gate delay.



internal capacitance C*	X	Y	Z	delay	ratio
if pre-discharged	0→1	0→1	1→0	84.3	1.00
if pre-charged	0→1	0→1	1→0	90.3	1.07

Figure 6: Various phenomena associated with MIS-TNC [5]

Figure 6, shows the various delay phenomena associated with MIS-TNC for 180 nm [5], we observed the same for our experiments with 65nm CMOS technology.

D. *Timing functions*

Given arrival times and transition times at a gate's inputs, and the initial state of internal capacitances, we compute timing functions for gate delays and output transition times [4][5]. The output arrival time and transition time is computed from the above data in our advanced timing analyzer - ETA (more in Section III.I).

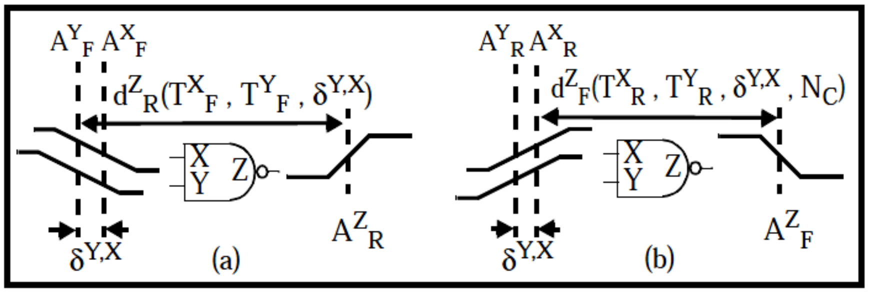


Figure 7: (a) Rise delay function and (b) Fall delay function

Figure 7 shows the delay timing functions for a 2-input NAND gate where all inputs of the gate have either steady non-controlling values or transitions in one direction. Similarly output transition time (rise and fall) functions can be obtained. Here, N_c represents the number of internal capacitances which are precharged to $V_{dd} - V_{th}$ and $\delta = A_Y - A_X$ is the skew. Gate delay can now be represented by the following timing functions:

- MIS-TC: Rise delay function d^Z_R (T^X_F, T^Y_F, $\delta^{Y,X}$).
- MIS-TNC: Fall delay function d^Z_F (T^X_R, T^Y_R, $\delta^{Y,X}$, N_c).

The timing functions can then be approximated into a piecewise linear model using empirical equations arrived at by using curve fitting as shown in Figure 8 where the accuracy of the model can be traded off with the complexity for development and the usage in the subsequent timing analysis. Note that any approximation we use to reduce complexity ensures that our model bounds actual delay in silicon.

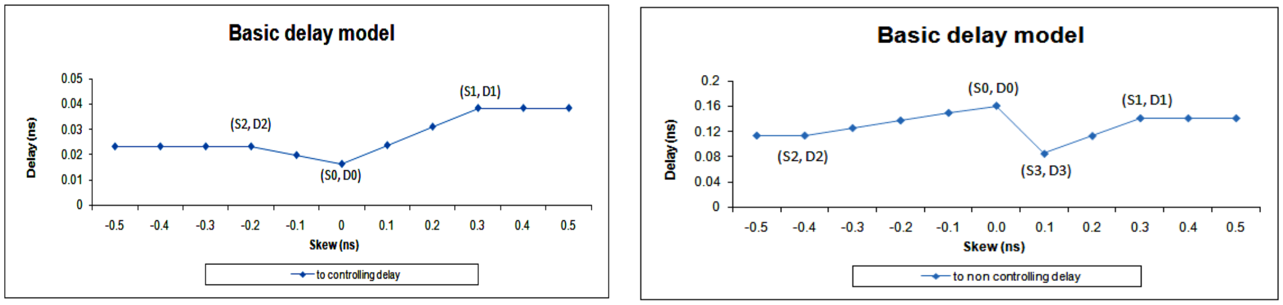


Figure 8: Piecewise linear approximation for near-simultaneous transitions (Basic delay model)

E. *Incorporating variability and bounding approximations*

A single curve combining the two cases (shown in Figure 5) cannot capture all inaccuracies and variations. Hence, we capture the inaccuracies and variations in these delay parameters using bounding approximations. We consider process variations in terms of the parameters of the devices in the gates, such as V_{th}, L_{eff} etc. Using the values of variations (from a foundry that fabricates chips in 65nm technology) for about 50 circuit parameters (including the major ones of V_{th}, L_{eff} etc), we perform Monte Carlo simulations to obtain the two envelopes to bound the constellation of points representing the gate delay under variability, as represented by the two outer envelopes in Figure 9.

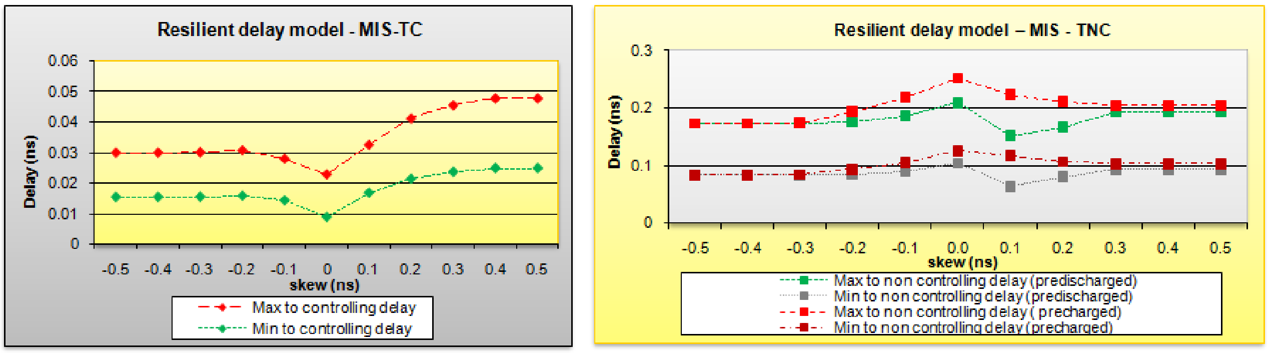
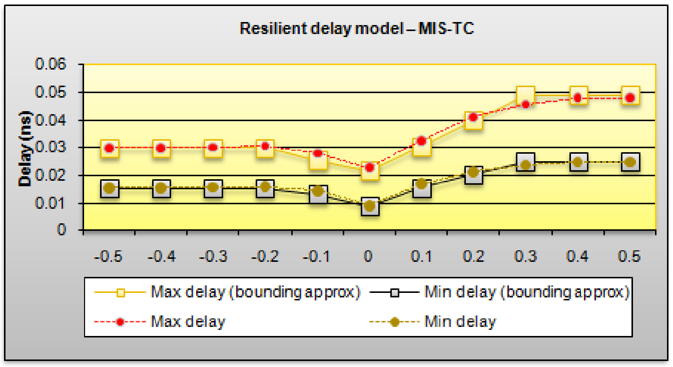


Figure 9: Delay vs skew curve for near-simultaneous transitions (Resilient delay model – simulations result)

The relationship of Figure 9 can be easily represented as a pair of three/four point piecewise linear approximations (curves with big squares at each data point in Figure 10). The two envelopes obtained, represent the bounded approximations for the resilient delay model.

The tightness of the bound defines the cost-benefit of the subsequent steps of timing analysis, path selection, and vector generation. The relationship between output transition times and skew for both MIS-TC and MIS-TNC are derived in a similar manner. Also, for each timing function in [4][5] we now have two sets of functions corresponding to the upper and lower bounds.



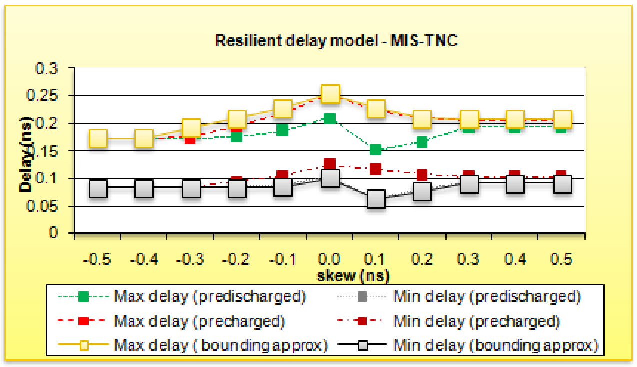
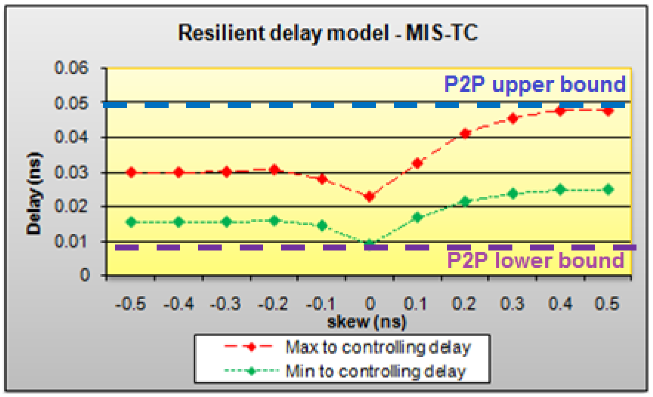


Figure 10: Resilient delay model – 3,4 point linear bounding

We would also like to bring to attention to the fact that though characterization of delay model is originally done for 11 skew points from -500ps to +500ps in steps of 100ps, we only store the 3/4 skew points corresponding to the three/four point approximation and hence do not increase the characterization effort significantly (see Section III.F).



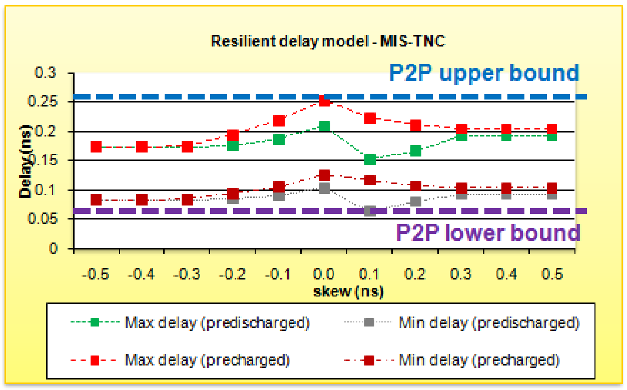


Figure 11: Resilient delay model – pin to pin bounding

The basic pin-to-pin delay model looks like a step function with a step at zero skew, where the size of step captures the difference between the pin-to-pin delays for the two inputs. Another way to bound the resilient delay model will be to use the simple pin-to-pin delay model to derive the bounds (Figure 11). It should be noted that in cases such as those for MIS-TC and MIS-TNC the step function of pin-to-pin delay model will reduce to a flat line just like the traditional corner based worst-case delay model.

Table III shows the superiority and tightness of our bounding approximations with respect to the simple pin-to-pin bounding (over all vectors). Here the overestimation of delay by an approach indicates that the delay reported by the approach is greater than the actual maximum delay or less than the actual minimum delay.

TABLE III: Tightness of bounding approximations

Type of bounds	**Overestimate max error (%)**			
	MIS-TC		MIS-TNC	
	Max delay	Min delay	Max delay	Min delay
Pin-to-pin	94.2%	64%	45.4%	50.4%
3/4 point	4.4%	5.3%	5.1%	5.5%

Our proposed 3/4 point bounds are automatically tighter (maximum error of about 5% compared to 90% for the bounds based on industry prevalent pin to pin models) because of our objective of emulating the reality as closely as possible at low complexity. Since the resilient delay model will be used for post-silicon tasks, it is imperative that it must capture the worst case with minimum looseness so that the subsequent post-silicon tasks do not explode in complexity.

F. Complexity trade-offs

In Table IV we compare our VRM based approach with the CSM based approach of [16] (which captures most of the known phenomena associated with gate delays) for characterization and usage complexities. The values in Table IV are for the delay of a single two input NAND gate characterized for 5 values of input slews and 5 values of output load. Suppose, for our approach we characterized for 7 values of skews for MIS (both MIS-TC and MIS-TNC) and 2 values of state of internal capacitance for MIS-TNC. Thus total transient simulations for our approach will be 5*4*7*(2+1) = 420. The corresponding value for CSM based approach [16] with input waveform sampled at 7 points will be $4*7^4$= 9,604 transient and 9,604 dc simulations. We consider about 10,000 Monte Carlo simulations [6] for characterization with full variability, so the corresponding characterization effort for our approach with variability is $4.2*10^6$. The statistical approach of CSM [20] requires the sensitivity of each parameter of variation to be calculated separately. Given, that for the library under consideration we have about

50 such parameters we arrive at the number of simulations to be $9.6*10^9$ (dc + transient).

We store curves for MIS_TC and MIS_TNC as three and four point approximations respectively. Hence, in a Table form we need to store 140 points which can be further optimized in an equation format by using curve fitting techniques to 7 points only. For variability, we store the two bounding curves for each set of input slew and output load hence the corresponding number of points for table and equation forms are 280 and 40 respectively. On the other hand, the CSM model needs to store output current, Miller capacitances, output and input capacitances as functions of input and output voltages which require about 48,000 points to be stored in table form. This can be further reduced to equation form at the cost of some accuracy (about 8%) by about 90% using compression techniques proposed in [37]. The statistical model [20] of CSM will require individual sensitivity tables and thus the storage complexity increases.

TABLE IV: Complexity trade-offs: VRM V/s CSM for a 2 input NAND Gate

	Nominal		**Full Variability**	
	VRM	**CSM**	**VRM**	**CSM**
Characterization complexity (number of simulations performed)				
Transient simulations	420	9,604	4.2×10^6	4.8×10^9
DC simulations	0	9,604	0	4.8×10^9
Characterization complexity (storage size)				
Table form	140	48,020	280	2.4×10^6
Equation form	7	4,800	14	2.4×10^5
Usage in timing analysis complexity (number of operations performed)				
Delay calculation	15	350	15	17,000

Our current approach for characterization requires about 7 simulations for various skews for each two input gate with a single load. We can let go of some accuracy to reduce simulation effort by considering a bound on near-simultaneous range as a function of maximum pin to pin delay. For example consider the near simultaneous region bounded by twice the pin to pin delay on either side of zero skew between near simultaneous transitions. In such a scenario the characterization effort for MIS just requires storage at one additional point – the zero skew and simulation runs reduce from 420 to 180. Also, we have observed the two bounding envelopes to be highly correlated and this property can be used to reduce storage complexity where only one envelope is stored and the other is derived from the first one.

Characterization is a non-recurring cost but timing analysis runtime is a huge recurring cost for test generation. Hence, we evaluated the complexity of delay calculation for a fully specified vector by counting the number of floating operations performed. We assign a weight of 1,2 and 5 to addition, multiplication and division respectively. Bottom row of Table IV clearly indicates that our approach though with diminished accuracy, is much more runtime efficient than CSM methods [16][20]. Later in Section V, we will show that timing analysis for unspecified and partially specified vectors using CSM is highly impractical.

TABLE V: Complexity trade-offs: Our approach v/s pin-to-pin bounding

	Nominal		**Full Variability**	
	3/4 point	**Pin-to-pin**	**3/4 point**	**Pin-to-pin**
Characterization complexity (number of simulations performed)				
Transient simulations	420	420	4.2×10^6	4.2×10^6
Characterization complexity (storage size)				
Equation form	7	1	14	2
Usage in timing analysis complexity (number of operations performed)				
Delay calculation	15	1	15	1

Table V shows the complexity trade-off of our approach of 3/4 point approximation v/s pin-to-pin bounding (based on pin-to-pin delay model prevalent in industry standard STA tools). As can be seen here, characterization effort in terms of simulations performed is identical to our approach (as all the known effects must be characterized for completeness). Though storage and timing analysis complexity reduces drastically, the inherent looseness in bounds (50% to 90% for a single gate) calculated from this approach (see Table III) and the enormous increase in complexity of post-silicon tasks (see Table XII) renders pin-to-pin bounding (also, known as worst case delay model) impractical for post-silicon delay related tasks.

G. Extended Model

The proposed resilient delay model can handle different numbers of inputs, input positions and can be extended to handle more than two simultaneous transitions using the approaches from [5]. The extended delay model though more accurate, needs more cases to be enumerated and the corresponding characterization effort increases. Also, the timing analysis framework needs to consider more timing cases and thus the runtime complexity explodes. A workaround is to decompose the gates in the circuit as 2-input gates for delay calculation for post-silicon delay related tasks.

H. Application of delay model

The aforesaid delay model can be used for the following three pre-silicon tasks targeting post-silicon delay related activities (not central to this paper, so refer to [27][28][29] for more details).

1) Enhanced Timing Analysis

Given the arrival and transition times at a gate's inputs, we calculate the corresponding quantities at the gate's outputs. Figure 12 shows the possible input combinations for a rising/falling transition at output.

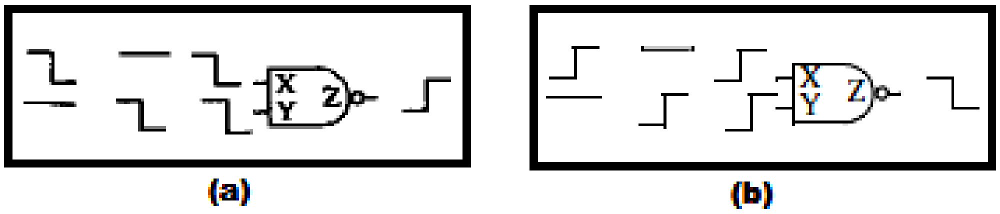


Figure 12: Possible input combinations for (a) output rising transition (b) output falling transition.

We enhanced the approach in [27] for both MIS-TC and MIS-TNC, where by using the bitonic relationship between delay and input transition time and exploring the possible transition times within the min-max range in the input transition time vs. delay curves, one arrives at the equations for output arrival times (see Figure 13) and output transition times.

$$
\boxed{
\begin{aligned}
A^Z_{RS} &= \min[A^X_{FS}, A^Y_{FS}] + min_{\beta\gamma \in \{S,L\}}[d^Z_{Rmin}(T^X_{F\beta}, T^Y_{F\gamma}, A^Y_{FS} - A^X_{FS})] \\
A^Z_{RL} &= \max[A^X_{FL} + d^{Z,X}_{Rmax}(T^{X*}_F), A^Y_{FL} + d^{Z,Y}_{Rmax}(T^{Y*}_F)] \\
A^Z_{FS} &= \min\Big[A^X_{RS} + \min\{d^{Z,X}_{Fmin}(T^X_{RS}), d^{Z,X}_{Fmin}(T^X_{RL})\}, A^Y_{RS} + \min\{d^{Z,Y}_{Fmin}(T^Y_{RS}), d^{Z,Y}_{Fmin}(T^Y_{RL})\}, \\
&\qquad \max[A^X_{RS}, A^Y_{RS}] + min_{\beta\gamma \in \{S,L\}}[d^Z_{Fmin}(T^X_{R\beta}, T^Y_{R\gamma}, A^Y_{RS} - A^X_{RS}, N_c = 0)]\Big] \\
A^Z_{FL} &= \max[A^X_{RL}, A^Y_{RL}] + max_{\beta\gamma \in \{S,L\}}[d^Z_{Fmax}(T^X_{R\beta}, T^Y_{R\gamma}, A^Y_{RL} - A^X_{RL}, N_c = 1)]
\end{aligned}
}
$$

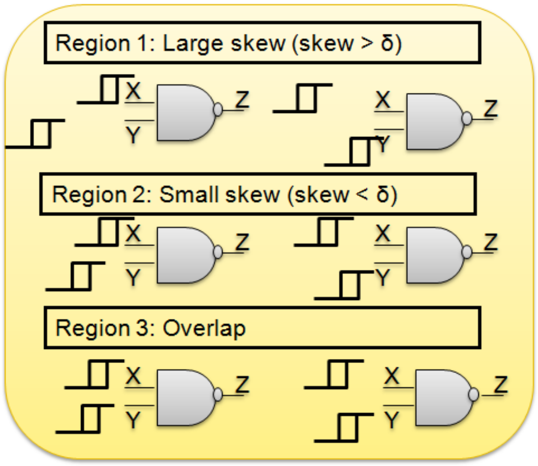
Figure 13: Calculation of output arrival times in ETA using resilient delay model.

2) Path selection

We enhanced the approach in [28] that identifies a set of paths that is guaranteed to include all paths that may potentially cause a timing error if the accumulated values of additional delays along circuit paths is upper bounded by a desired limit (Δ), works with upper and lower bounds given by our resilient delay model and also checks for both functional sensitization and high delay excitation.

3) *Vector generation approach*

In [35] impact of multiple input switching for vector generation under process variation is considered but the generated tests do not guarantee to invoke the worst case delay. This motivated us to derive a framework to accurately estimate the delay of a fabricated chip post-silicon [29].



Region 1: Large skew (skew > δ)
X
Y
Z
Region 2: Small skew (skew < δ)
Region 3: Overlap

Region	Timing case
1	1 and 5
2	2 and 4
3	3

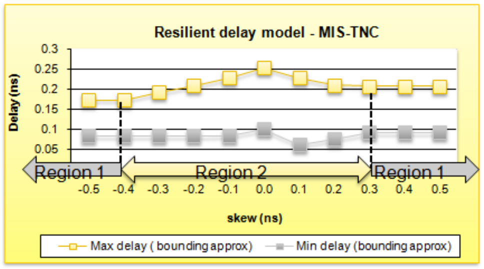


Figure 14: Taxonomy of timing cases using resilient delay model [29]

In [29] a new approach to generate vectors for post silicon delay characterization using the proposed resilient delay model is presented. The method generates vectors that are guaranteed to excite the worst-case delays of fabricated chips without introducing any pessimism by intelligently dividing the delay model to various timing ranges (see Figure 14) and innovatively exploiting the effect of MIS-TC and MIS-TNC on the gate delay in these timing ranges.

IV. EXPERIMENTAL RESULTS

We applied our approach to combinational parts of ISCAS89 benchmark circuits (see Table VI) using an Intel Core 2 Duo 2.2 GHz machine. All gates in the benchmark circuits are assumed to use minimum-size transistors, and a 65nm CMOS technology is used. Our experiments used our new resilient delay model for both to-controlling (MIS-TC) and to-non-controlling (MIS-TNC) transitions. Using the approach described in Section III we characterize all the basic gates (NAND, NOR, AND, OR, NOT and BUF).

TABLE VI: ISCAS89 Benchmark Circuits

Benchmark	PIs	POs	# of logical Paths
s298	17	20	462
s444	24	27	1,070
s953	45	52	2,312
s1196	32	32	6,196
s5378	214	228	27,084
s9234	247	250	489,708

A. *Experiments on accuracy of timing analysis*

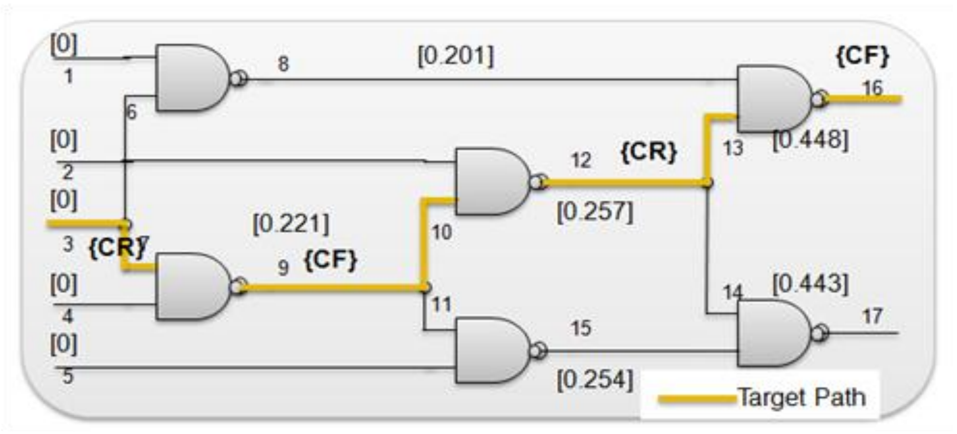


Figure 15: The benchmark c17

We did a preliminary experiment on ISCAS85 benchmark c17 (Figure 15). First, for each primary output (lines 16 and 17) we identify the input cone. We perform cone-exhaustive simulations on cone A (line # 16) and cone B (line # 17) via detailed circuit simulations using Cadence-Spectre to identify the worst case delay invoking vector and compare it with the timing estimates obtained by our approach. We repeated the experiment for the cones corresponding to the top three critical paths (cone 1 for all the three paths) for s298 also (see Table VII).

In Table VII, column 3 shows the number of circuit lines in the fan-in cone, column 5 shows the max delay calculated at the primary output under consideration using circuit simulations. Column 6 (timing simulation) and column 8 (timing analysis), shows the same for our method; column 9 and column 10 indicate weather our approach successfully bounds the delay values obtained from circuit simulations and for how many lines it can do so. Column 7 and column 11 indicates the tightness of the bounds obtained by our method with respect to the circuit simulation results. The results indicate that our proposed analysis approach does bound all the lines in the nominal case with an error of about 1% for c17 and 3% for s298, whereas the corresponding figures for our proposed simulation based approach reduces to 0.5% (c17) and 1% (s298) . The error can be associated with the looseness associated with the timing ranges calculated by static timing analysis and the looseness of the bounding approximations.

TABLE VII: Analysis vs Simulation for smaller ISCAS Benchmarks

	PO	# lines		Max delay (SPECTRE Simulations)	Our approach (Simulations)		Our approach (Analysis)			
					Max delay	Tightness of bounds (max)	Max delay	True Bound	# lines bounded	Tightness of bounds (max)
Analysis for c17										
ConeA	16	12	**Nominal**	0.444ns	0.445ns	0.02%	0.448ns	Y	12	0.09%
			Full variability	0.626ns	0.662ns	5.75%	0.676ns	Y	12	7.98%
Cone B	17	12	**Nominal**	0.443ns	0.445ns	0.45%	0.448ns	Y	12	1.12%
			Full variability	0.607ns	0.632ns	4.11%	0.660ns	Y	12	8.73%
Analysis for s298										
Cone 1	250	65	**Nominal**	0.626ns	0.634ns	1.2%	0.639ns	Y	65	2.08%
			Full variability	0.807ns	0.856ns	6.07%	0.894ns	Y	65	10.78%
Cone 1	250	65	**Nominal**	0.625ns	0.630ns	0.80%	0.639ns	Y	65	2.23%
			Full variability	0.807ns	0.856ns	6.07%	0.894ns	Y	65	10.78%
Cone 1	250	65	**Nominal**	0.624ns	0.631ns	0.93%	0.639ns	Y	65	2.40%
			Full variability	0.806ns	0.856ns	6.20%	0.894ns	Y	65	10.91%

In another set of experiments we perform sufficiently large number of Monte Carlo simulations (with full global variability) on benchmarks c17 and s298 using the vector identified in the previous step. Note that the max delay we are reporting here is actually the maximum arrival time at the output which can serve as a tight upper bound on the worst case delay. Also for the full global variability, we do have 100% bounding but the bounds are loose than the nominal case. Experiments with variability results in an error of about 9% (c17) and 11% (s298), whereas the corresponding figures for our proposed simulation based approach reduces to 6% (c17) and 7% (s298) . One explanation for this behavior is that we did same number of Monte Carlo runs (say *k*, where *k* is sufficiently large) for both the cases (basic gates and complete circuit). So the *k* runs for the circuit don't cover the process space that has been covered during *k* runs for the basic gates during the characterization step. We also compared the accuracy of our approach for c17 and s298 with respect to SPECTRE using our resilient delay model with 3/4 point bounding and with pin-to-pin delay bounding. Table VIII shows our resilient delay model with 3/4 point bounding gives way better results than delay model based on pin-to-pin (used by existing STA tools [33]) bounding.

TABLE VIII: Accuracy of ETA

Benchmark	ETA Accuracy – Tightness of bounds (max) (%)			
	Nominal		Full Variability	
	3/4 bounding	p2p bounding	3/4 bounding	p2p bounding
c17	1%	2.5%	9%	11%
s298	3%	11%	11%	28%

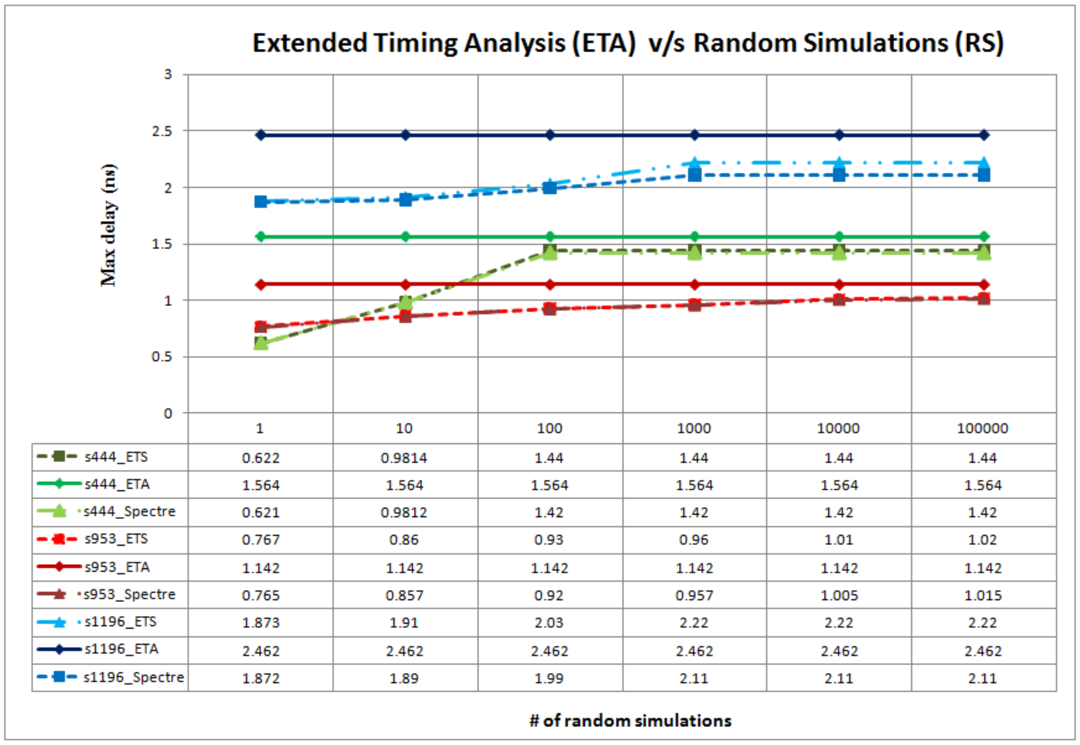
Cone exhaustive simulation even for medium size benchmark is a tedious task and can take up to several days of simulation time. Hence for such circuits, we simulated randomly generated 10,000 vectors using Spectre and ETS (our timing simulator) and compared them to the result calculated by our ETA (Table IX). As can be seen from Table IX the max delay for nominal case (nom) for s444, s953 and s1196 calculated by our ETA (at negligible computation cost) is offset by about 10.14%, 12.51% and 16.68% respectively with respect to 10,000 random Spectre simulations (which requires large amount of simulation time). Also, the results reported by our ETS (at considerable computation cost) are much more accurate with error being only 1.4%, 0.5% and 5.2% for s444, s953 and s1196 respectively.

TABLE IX: Analysis for medium size ISCAS benchmarks

Benchmark	Max delay reported (ns)			Accuracy (%)	
	Random Spectre simulations	Random ETS simulations	ETA (Analysis)	ETA	ETS
s444	1.42	1.44	1.564	10.14%	1.4%
s953	1.015	1.02	1.142	12.51%	0.5%
s1196	2.11	2.22	2.462	16.68%	5.2%

Figure 16 shows the results for our random simulations. The trend of the curves does indicate that the random simulations and timing analysis can potentially converge for sufficiently large number of simulations. It is important to note that in these experiments the looseness in probably due to the fact that the vector that can invoke the worst case delay might not be applied during the random simulations.

When we use our approach for generating vectors for validation to identify vectors [29] and analyze them using our approach on a vector-by-vector basis, the nominal worst case delay observed for s973 becomes 1.132ns. Compared to this, the looseness of our static approach reduces from 12.5% (from random simulations) to 8.83% (see Table X), demonstrating that our static approach is indeed tight and the looseness in the above figures can be further attributed to the inability of randomly generated vectors to invoke worst-case delays. We would like to restate the fact that this looseness of around 10% for medium benchmark circuits such as s973 incorporates the uncertainty in vector space (unspecified) which none of the other timing analysis approaches [17][22][34] have accounted for while reporting their results.



	1	10	100	1000	10000	100000
s444_ETS	0.622	0.9814	1.44	1.44	1.44	1.44
s444_ETA	1.564	1.564	1.564	1.564	1.564	1.564
s444_Spectre	0.621	0.9812	1.42	1.42	1.42	1.42
s953_ETS	0.767	0.86	0.93	0.96	1.01	1.02
s953_ETA	1.142	1.142	1.142	1.142	1.142	1.142
s953_Spectre	0.765	0.857	0.92	0.957	1.005	1.015
s1196_ETS	1.873	1.91	2.03	2.22	2.22	2.22
s1196_ETA	2.462	2.462	2.462	2.462	2.462	2.462
s1196_Spectre	1.872	1.89	1.99	2.11	2.11	2.11

Figure 16: ETA V/s Random simulations

Monte Carlo simulations using circuit simulators for these medium size circuits are very time consuming, and given the number of process parameters varied for the 65nm industrial library provided to us, such simulations will take days. Hence for experiments with variability on these benchmarks we report the max delay obtained by random simulations and compare them to the result calculated by our ETA (see Table X). The inaccuracy of ETA increases for the experiments with variability (from 10.9 % to 18.61% for s1196 w.r.t. random ETS simulations). This inaccuracy can be attributed to Monte Carlo runs on the full circuit not covering the complete process space that is covered during characterization of gates.

TABLE X: ETA V/s Random Simulations V/s MDS simulations [29] for medium ISCAS benchmarks

Benchmark	ETA Accuracy – Tightness of bounds (max) (%)			
	Nominal		Full global variability	
	w.r.t. Random ETS Simulations (10,000)	w.r.t. MDS Simulations [29]	w.r.t. Random ETS Simulations (10,000)	w.r.t. MDS Simulations [29]
s444	8.61%	4.61%	10.5%	5.2%
s953	11.96%	8.83%	15.42%	9.4%
s1196	10.9%	4.23%	18.61%	11.2%

We performed similar experiments with bigger benchmarks s5378 and s9234 and report the results in Table XI. Even for s9234 the looseness of ETA is around 18% (compared to MDS simulations) which is a reasonable accuracy considering the fact that ETA ran only once in a vector unaware manner to achieve this.

TABLE XI: ETA V/s Random Simulations V/s MDS simulations [29] for big ISCAS benchmarks

Benchmark	**ETA Accuracy – Tightness of bounds (max) (%)**			
	Nominal		**Full global variability**	
	w.r.t. Random ETS Simulations (10,000)	**w.r.t. MDS Simulations [29]**	**w.r.t. Random ETS Simulations (10,000)**	**w.r.t. MDS Simulations [29]**
s5378	19%	12%	25%	14%
s9234	33%	18%	42%	21%

Since the proposed delay model will be used primarily for post-silicon delay related tasks and not design analysis, the delay will be measured on the fabricated chip and not estimated by the loose ETA [28]. The bounding approximation will definitely contribute to the looseness of the estimation but the delay measured by applying vectors generated by our approach [29] will be the actual delay with *zero* pessimism.

Our framework can work with fully unspecified, partially specified, and fully specified vectors and hence is suitable for post-silicon delay related tasks such as timing characterization and delay validation. We demonstrate this on s298 by specifying certain bits in a primary input sequence (test-vector) and report the results in Table XII. The first row reports the result of ETA with fully unspecified vectors (equivalent of static timing analysis), whereas the last row does the same for a fully specified vector used for maximum delay invocation [29]. The intermediate rows report (the average over ten random choices evaluated for each case of bit specification) the ETA results with partially specified vectors.

TABLE XII: Analysis on s298 with partially specified vectors

# of bits specified	**Nominal**		**Full global variability**	
	A_min (ns)	**A_max (ns)**	**A_min (ns)**	**A_max (ns)**
0	0.290	0.639	0.207	0.894
2	0.289	0.638	0.206	0.882
4	0.288	0.638	0.206	0.878
8	0.276	0.635	0.174	0.873
All	0.276	0.634	0.0084	0.856

It is important to note that a vector unaware approach (using fully unspecified vectors) can overestimate actual maximum arrival times (around 2.08% and 10.78% for the nominal and full variability cases respectively). The corresponding figures for a vector aware simulation based approach are 1.2% and 6.07% (see Table VII). Also, as expected the range for fully unspecified vector subsumes that for the partially specified vector, which in turn subsumes that for the fully specified vector.

Table XIII provides an estimate of the run-time complexity associated with performing timing analysis for partially specified vectors using CSM based approaches [22][34]. As evident from the numbers in fourth column that CSM based approaches even after being superior for pre-silicon tasks are rendered almost impractical for post-silicon tasks.

TABLE XIII: Runtime Analysis: Our approach V/s CSM based approaches [22][34]

BENCHMARK	**# of bits specified**	**Number of simulation runs required**	
		Our approach	**CSM based approaches [22][34]**
s298	**0**	1	1.71×10^{10}
	10	1	1.67×10^{7}
	All	1	1
s953	**0**	1	1.23×10^{27}
	10	1	1.2×10^{24}
	All	1	1
s9234	**0**	1	5.11×10^{148}
	10	1	4.99×10^{145}
	All	1	1

B. Experiments on path selection and vector generation

In our second set of experiments, we characterize the delay for each gate using full global variability and generated vectors for validation using our delay validation framework [29]. In these experiments, since the variations are incorporated in the delay models, the timing threshold $\Delta = 0.02$ [32] captures only modeling errors.

Table XIV shows the comparison of our delay model with the worst case delay model on various delay marginality validation metrics such as selected path set, generated vector set and test generation runtime (given in CPU clocks provided by the clock() function in C, an approximation of processor time). Results clearly indicate the superiority of our proposed delay model over the bounded pin-to-pin delay model.

Also, for s953 even though the vector space has increased by about 700%, the test generation time has increased by about 25% only. This can be attributed to the fact that a simpler delay model might gain in the delay calculation phase, but the inherent looseness in bounds will increase the search space leading to increase in the size of the selected path set and the vector-space that is searched. In [29] we have shown that the vectors generated using our resilient delay model can invoke more delay (up to 15% in some cases) compared to robust test vectors.

TABLE XIV: Analysis for delay validation

Benchmark	**Resilient delay model**			**Bounded P2P delay model**		
	Paths	**Vectors**	**CPU clocks**	**Paths**	**Vectors**	**CPU clocks**
s298	2	64	4,179	11	74	6,364
s953	2	2	83,467	6	12	97,264
s1196	18	936	213,476	25	7,552	354,4468
s9234	5,346	4.35×10^5	2.67×10^8	18,764	8.15×10^{10}	1.17×10^9

Also the validation vector set, though large, is practical because post-silicon validation is performed for a small sample of chips selected from first-silicon batch (unlike delay testing, which must be performed on every fabricated chip copy). Note that the test vector spaces generated by our approach can be further refined using test compaction methods [38].

V. CONCLUSION

Experimental results demonstrate that our new resilient delay model based on bounding approximations captures the effect of variability and yet generates tight bounds. It can also tighten these bounds using available logic values at any circuit lines and thus is suitable for post silicon tasks. It is also a dramatic improvement over the traditional worst-case delay model in the terms of paths selected and vectors generated for validation. Furthermore our philosophy of bounding approximations compensates for lack of exact knowledge and inaccuracies in delay parameters. We have also successfully developed a framework based on our resilient delay model that uses our new timing dependent conditions and a new vector generation approach to generate a practically useful set of vectors for post silicon delay characterization, including delay marginality validation and speed binning [29].

We are currently developing approaches to exploit chip sampling using the knowledge of die-to-die vs. on-die variability to dramatically reduce the number of vectors required for delay characterization. We are also conducting experiments on actual silicon to further validate our approach, for validation as well as speed binning. We also intend to extend our approach to develop models for accurate analysis of other lower order delay effects, especially crosstalk and ground bounce.

ACKNOWLEDGEMENT

This research is supported by Intel Corporation.

References

[1]B. D. Cory, R. Kapur, and B. Underwood, “Speed Binning with Path Delay Test in 150-nm technology,” In *IEEE Design & Test of Computers,* Vol.20, Sept-Oct 2003, Issue 4, pp. 41-45.

[2]J. Zeng, M. Abadir, G. Vandling, et al., “On correlating structural tests with functional tests for speed binning of high performance design,” In *Proc. of International Test Conf.*, 2004, pp. 31-37.

[3]L. C. Chen, P. Dickinson, P. Dahlgren, et al., “Using Transition Test to Understand Timing Behavior of Logic Circuits on UltraSPARC^{TM}T2 Family,” In *International Test Conf.*, 2009, pp. 1-10.

[4]L. C. Chen, S. K. Gupta, and M. A. Breuer, “A New Gate Delay Model for Simultaneous Switching and Its Applications,” In *Proc. Design Automation Conf.*, 2001, pp. 289-294.

[5]L. C, Chen, S. K. Gupta, and M. A. Breuer, “Gate Delay Modeling for Multiple to non-controlling stimuli” – Unpublished.

[6]K. Bernstein et al., “High- Performance CMOS variability in the 65-nm regime and beyond,” In *IBM Journal of Research and Development*, Vol 50, Issue 4.5, 2006, pp. 433- 449.

[7]S. Nassif. “Delay Variability: Sources, Impact and Trends,” In *Proc. Solid-State Circuits Conf.*, 2000, pp. 368-369.

[8]International Technology Roadmap for semiconductors – www.itrs.net

[9]I. Keller, K. H. Tam, and V. Kariat, “Challenges in Gate level Modeling for Delay and SI at 65nm and below,” In *Proc. Design Automation Conf.*, 2008, pp. 468-473.

[10] X. Wang, A. Kasnavi, and H. Levy, "An efficient method for fast delay *and SI calculation using current source mode*ls," In *Proc. International Symp. On Quality Electronic Design*, 2008, pp. 58-61.

[11] J. F. Croix and D. F. Wang, "Blade and Razor: Cell and Inte*rconnect Delay Analysis Using Current-Based Models,"* In *Proc. Design Automation Conf.*, 2003, pp. 386-389.

[12] R. Goyal and N. Kumar, "Current Based Delay Models: A Must for Nanometer Timing"- *technical report CADENCE* – unpublished.

[13] J. Shin et al., "A Gate Delay model considering temporal proximity of multiple input switching," *In ISOCC*, 2009, pp. 577-580.

[14] S. Tsai and C. Huang, "A False-Path Aware Formal Static Timing Analyzer Considering Simultaneous Input Transitions," In *Proc. Design Automation Conf.*, 2009, pp. 25-30.

[15] D. Tadesse et al., "Accurate Timing Analysis using SAT and Pattern-Dependent Delay Models," In *Design, Automation & Test in Europe Conf.*, 2007, pp. 1-6.

[16] B. Amelifard et al., "A Current Source Model for CMOS logic cells considering multiple input switching and stack effect," In *Design, Automation & Test in Europe Conf.* , 2008, pp. 568-573.

[17] C. Amin et al, "A Multi-port Current Source Model for Multiple-Input Switching Effects in CMOS Library Cells," In *Proc. Design Automation Conf.*, 2006, pp. 247- 252.

[18] T. W. Chiang, C. Y. R. Chen, and W. Y. Chen, "An Efficient Gate Delay Model for VLSI Design," In *Intl. Conf. On Computer Design*, 2007, pp. 450-455.

[19] D. Blaauw et al., "Statistical Timing Analysis: From basic principles to state of the art," In *IEEE Trans. On Computer Aided Design of Integrated Circuits and Systems*, 2008, Vol. 7, Issue 4, pp. 589-607.

[20] H. Fatemi, S. Nazarian, and M. Pedram, "Statistical Logic Cell Delay Analysis Using a Current-based Model," In *Proc. Design Automation Conf.*, 2006, pp. 253-256.

[21] A.Goel and S. Vrudhula, "Statistical waveform and current source based standard cell models for accurate timing analysis," In *Proc. Design Automation Conf.*, 2008, pp. 227-230.

[22] V.Veetil, D. Sylvester, and D. Blauuw, “Fast and Accurate Waveform Analysis with Current Source Models,” In *Proc. International Symp. On Quality Electronic Design*, 2008, pp. 53-56.

[23] S. Yanamanamanda et al., “Uncertainty modeling of Gate Delay considering multiple input switching,” In *Proc. International Symp. On Circuits and Systems*, 2005, Vol. 3, pp. 2457-2460.

[24] J. Sridharan and T. Chen, "Gate Delay Modeling with Multiple Input Switching for Static (Statistical) Timing Analysis," in *Proc. VLSI Design*, 2006, pp.323-328.

[25] A. Agarwal, F. Dartu and D. Blauuw, “Statistical gate delay model considering Multiple Input Switching,” In *Proc. Design Automation Conf.*, 2004, pp. 658-663.

[26] S. Ganapathy et al., “Circuit propagation delay estimation through multivariate regres*sion based modeling under spatio- tempora*l variability,” In *Design, Automation & Test in Europe Conf.* , 2010, pp. 417-422.

[27] L. C. Chen, S. K. Gupta, and M. A. Breuer, “A New Framework for Static Timing Analysis, Incremental Timing Refinement, and Timing Simulation,” *Asian Test Symp.*, 2000, pp. 102-107.

[28] I. D. Huang and S. K. Gupta, "On Generating Vectors That Invoke High Circuit Delays - Delay Testing and Dynamic Timing Analysis," In *Proc. Asian Test Symp.* , 2007, pp.485-492

[29] P. Das and S. K. Gupta, “On generating vectors for accurate post-silicon delay characterization,” In *Proc. Asian Test Symp.*, 2011, pp. 251 - 260.

[30] F. Dartu, N. Menezes, J. Qian, and L. T. Pillage, “A Gate Delay Model for High Speed CMOS Circuits”, In *Proc. Design Automation Conf.*, 1994, pp. 576-580.

[31] Y. Cao, P. Gupta, et al., “Design Sensitivities to variability: Extrapolations and assessments in nanometer VLSI,” *In IEEE International ASIC/SOC Conf.*, 2002, pp. 411 – 415.

[32] I.D. Huang and S. K. Gupta, “Selection of Paths for Delay Testing,” In *Proc. Asian Test Symp*., 2005, pp. 208 - 215.

[33] Synopsys Prime Time SI – http://www.synopsys.com.

[34] S. Gupta and S. S. Sapatnekar, “Compact Current Source Models for Timing Analysis under Temperature and Body Bias Variations,” In *IEEE Trans. On VLSI Systems*, 2011, pp. 1 - 14.

[35] S. H. Wu, S. Chakravarty, and L. C. Wang, "Impact of Multiple Input Switching on Delay Test under Process Variation," In *Proc. VLSI Test Symp.*, 2010, pp. 87 - 92.

[36] S. Hatami, "Gate Delay Modeling and Static Timing Analysis in ASIC Design considering Process Variations," PhD Dissertation, submitted to USC, EE-Systems, 2011.

[37] S. Hatami and M. Pedram, "Efficient representation, stratification, and compression of variational CSM library waveforms using robust principle component analysis," In *Design, Automation & Test in Europe Conf.*, 2010, pp. 1285-1290.

[38] Z. Wang, D. M. H. Walker, "Dynamic Compaction for High Quality Delay Test," In *Proc. VLSI Test Symp.*, 2008, pp. 243 - 248.

[39] J. B. Sulistyo and D. S. Ha, "A new Characterization Method for Delay and Power Dissipation of Standard Library Cells", In *IEEE Journal of VLSI Design*, 2002, pp.667 - 678.

[40] C.H-P. Wen et al., "On A Software Based Self-Test Methodology and Its Application", In *Proc. VLSI Test Symp.*, 2005, pp. 107-113.

[41] S. Natarajan, S. Patil, and S. Chakravarty, "Path Delay Fault Simulation on Large Industrial Designs", In *Proc. VLSI Test Symp.*, 2006, pp. 16-23.

[42] C.L. Su et al., "Reducing Power Consumption at the Control Path of High Performance Microprocessors", In *IEEE Design and Test of Computers*, 1994, pp.1-20.