

SI for Free: Machine Learning of Interconnect Coupling Delay and Transition Effects

Andrew B. Kahng^{†‡}, Mulong Luo[‡] and Siddhartha Nath[‡]

[†]ECE and [‡]CSE Departments, UC San Diego, La Jolla, CA 92093

{abk, muluo, sinath}@ucsd.edu

Abstract—In advanced technology nodes, incremental delay due to coupling is a serious concern. Design companies spend significant resources on static timing analysis (STA) tool licenses with signal integrity (SI) enabled. The runtime of the STA tools in SI mode is typically large due to complex algorithms and iterative calculation of timing windows to accurately determine aggressor and victim alignments, as well as delay and slew estimations. In this work, we develop machine learning-based predictors of timing in SI mode based on timing reports from non-SI mode. Timing analysis in non-SI mode is faster and the license costs can be several times less than those of SI mode. We determine electrical and logic structure parameters that affect the incremental arc delay/slew and path delay (i.e., the difference in arrival times at the clock pin of the launch flip-flop and the D pin of the capture flip-flop) in SI mode, and develop models that can predict these SI-aware delays. We report worst-case error of 7.0ps and average error of 0.7ps for our models to predict incremental transition time, worst-case error of 5.2ps and average error of 1.2ps for our models to predict incremental delay, and worst-case error of 8.2ps and average error of 1.7ps for our models to predict path delay, in 28nm FDSOI technology. We also demonstrate that our models are robust across designs and signoff constraints at a particular technology node.

Keywords: Signal integrity (SI), incremental transition time, incremental delay, SI-aware path delay, machine learning

I. INTRODUCTION

Accurate signoff timing analysis must be conducted using signal integrity (SI) mode in signoff timing tools. According to recent reports of the analyst firm Gary Smith EDA [11], EDA vendors such as Atrenta [17], Cadence [18], CLK Design Automation [19], Incentia Design Systems [20], Mentor Graphics [22] and Synopsys [25], provide STA (Static Timing Analysis) and SI analysis tools for use in IC design. The cost of one license of a timing tool with SI mode analysis enabled is typically several times the cost of a default (with no SI analysis capability) license. In addition, the runtimes of SI-aware timing analysis are significantly larger than those of non-SI analysis. Our own studies indicate the runtime for SI-aware timing analysis on the top-10K paths can be up to $3\times$ longer than the runtime of non-SI analysis on designs with $\sim 110K$ instances and $\sim 110K$ nets.

As would be expected, commercial signoff timing tools show significant differences between SI and non-SI modes when estimating arc delay of a stage as well as the accumulated arc delays in a path. We have studied SI and non-SI analyses with the same commercial timer, netlists, 28nm FDSOI libraries, and SPEF. For the non-SI analysis, we add twice the coupling capacitance to the ground capacitance to model worst-case Miller coupling [10]. Figure 1 shows that the path slack can differ by up to 81ps between SI and non-SI analyses.

In this work, we use machine learning techniques to estimate the incremental transition time, incremental delay due to SI, and SI-aware path delay from reports of a signoff timer that performs only non-SI analysis. Table I introduces the terminologies and notations

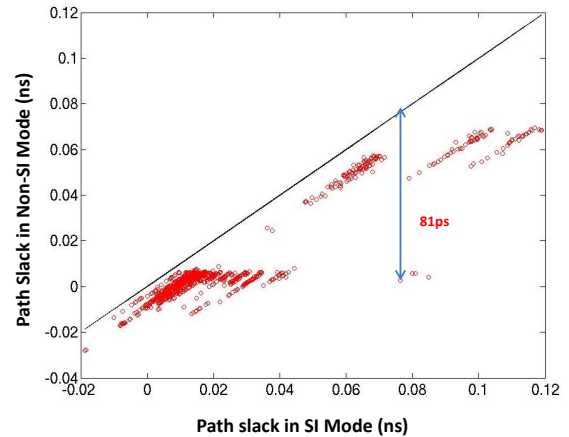


Fig. 1. Path slack divergence in SI and non-SI analyses with clock period 1.0ns, as reported by a commercial timer in 28nm FDSOI technology.

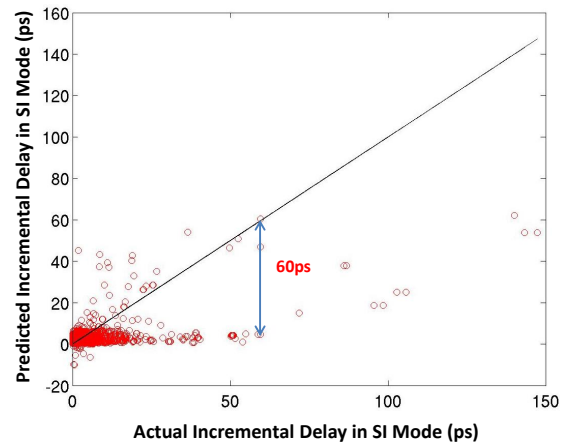


Fig. 2. Actual incremental delay in SI mode versus predictions with clock period of 1.0ns, using models of [4].

we use in our work. Previous work by Han et al. [4] provides methodologies to calibrate non-SI to non-SI, or SI to SI, but does not attempt our present mapping of non-SI to SI. This is the gap that the present work seeks to fill. Figure 2 shows that the prediction of incremental delay in SI mode can be inaccurate by up to 60ps when using the wire delay model in [4] and timing reports from non-SI analysis.

Multiple parameters ranging from electrical to logic structure such as coupling capacitance, the ratio of ground and coupling capacitance of an arc, clock period, the fanin cone stage of the arc, etc. all affect the divergence of transition times and delays between SI and non-SI analyses. Complex interactions among these parameters, along with black-box code in commercial signoff timers, only make the modeling problem more difficult. For example,

TABLE I
TERMINOLOGIES AND NOTATIONS.

Term	Definition
SI mode	Timing analysis performed by enabling signal integrity
Non-SI mode	Timing analysis performed by disabling signal integrity
C_c	Coupling capacitance of an arc
C_g	Ground capacitance of an arc
C_{tot}	Total capacitance of an arc
$r_{C_c, C_{tot}}$	Ratio of coupling to total capacitance of an arc
R_w	Resistance of an arc
ΔT_{si}	Delta transition (<i>DTran</i>) time of an arc due to coupling reported in timing analysis in SI mode
$T_{si'}$	Transition time of an arc without coupling reported in timing analysis in non-SI mode
ΔD_{si}	Incremental SI delay (<i>SI Incr Delay</i>) of an arc due to coupling reported in timing analysis in SI mode
$\Delta D_{si'}$	Incremental non-SI delay (<i>Non-SI Incr Delay</i>) of an arc without coupling reported in timing analysis in non-SI mode
Path delay	Difference in arrival times at the clock pin of the launch flip-flop and D pin of the capture flip-flop
P_{si}	SI path delay across all timing arcs reported in timing analysis in SI mode
$P_{si'}$	Non-SI path delay across all timing arcs reported in timing analysis in non-SI mode
ΔP_{si}	Difference between $P_{si'}$ and P_{si}
$f_{c, red}$	Miller coupling factor in non-SI mode, i.e., $C_c \times f_{c, red}$ is added to C_g
f_c	Coupling capacitance factor in SI mode, i.e., C_c is changed to $C_c \times f_c$
f_g	Ground capacitance factor in SI or non-SI mode, i.e., C_g is changed to $C_g \times f_g$
f_{R_w}	Resistance factor in SI or non-SI mode, i.e., R_w is changed to $R_w \times f_{R_w}$
S	Stage in which the arc appears
N_{stg}	Number of stages in the path in which arc appears
$r_{S, N_{stg}}$	Ratio of arc-stage to total #stages in path
$clkp$	Clock period
N_{aggr}	Number of aggressors for a victim net
A_r	Toggle rate of a net
$arr_{(min, max), (r, f), (a, v)}$	Minimum (resp. maximum) rise (resp. fall) arrival time of an aggressor (resp. a victim)
LE	Logical effort of the driver of a net

change in the clock period changes toggle rates of aggressor and victim nets by different amounts that can lead to change in aggressor and victim timing window alignment. Two phenomena are particularly challenging for analytical SI delay models.

Challenge 1. Path slack variation with clock period. Figure 3 shows the maximum delta of slack in a path with 32 stages between SI and non-SI analyses for an OpenCores [23] design *dec_viterbi* that is signed off at 1.0ns. The delta is 81ps when the clock period varies between 0.87ns and 1.3ns. However, when the clock period decreases below 0.87ns, the maximum delta in path slack increases non-monotonically and becomes 143ps at a clock period of 0.8ns. Figure 4 shows timing parameters related to SI and non-SI analyses for several nets and cells. The nets *n33458* and *n33452* shown in brown are responsible for large delta transition times and incremental delays in SI mode. We highlight these deltas and the impact to path slack using the blue box. The same path has a delta slack of 49ps when the clock period is 1.0ns, as shown in Figure 5. The path that has the maximum delta slack of 81ps at a clock period of 1.0ns continues to have the same value of delta slack at a clock period of 0.8ns, as shown in Figure 6.

Challenge 2. Arc delay and incremental transition time variation with ground and coupling capacitances. We illustrate non-intuitive impacts of varying ground and coupling capacitances of the victim net *n33452* on arc delay and incremental transition time respectively in Figures 7(a) and (b). When the ground capacitance is changed from 0.006pF to 0.0132pF, the incremental delay in non-

SI mode increases from 4ps to 6ps, whereas the incremental delay due to coupling changes from 115ps to 100ps while delta transition time changes from 133ps to 147ps. The incremental delay and delta transition time in SI mode are affected in non-intuitive ways by changing the ratio of ground-to-coupling capacitance.

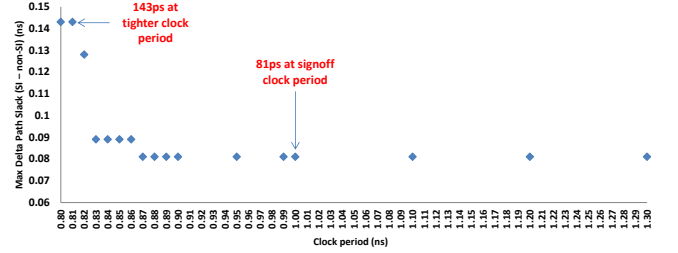


Fig. 3. Maximum path slack delta between SI and non-SI modes over the top-1000 setup-critical paths in a design signed off at 1.0ns. The delta increases from 81ps to 143ps as the clock period is reduced below 0.87ns.

Our contributions in this paper are summarized as follows.

- (1) We analyze multiple sources that cause timing divergence between SI and non-SI modes and provide new insights on electrical and logic structure parameters that affect incremental transition time, incremental delay and path delay in SI mode. Unlike [4], we demonstrate that several new parameters affect *SI Incr Delay* ΔD_{si} (as defined in Table I) of an arc in a timing path.
- (2) We develop new machine learning-based models for incremental transition time and delay due to SI, and compose these models to derive a new model for path delay that is different from [4].
- (3) The worst-case absolute errors in our modeling predictions of incremental transition time, incremental delay due to SI and SI-aware path delay are 7.0ps, 5.2ps and 8.2ps, respectively. We have developed and tested our models using timing reports of block implementations with 28nm FDSOI foundry libraries. Compared to the recent work of [4], we reduce worst-case error in prediction of incremental delay due to SI changes from 60ps to 5.2ps.

The remainder of the paper is organized as follows. In Section II, we review related works on studying correlations of timing reports/predictions between different tools/models with attention to SI effect. In Section III, we describe our methodology to select significant parameters and derive machine learning models for incremental delay and path delay in SI mode. In Section IV, we describe our experimental setup and present results. In Section V, we describe future works and conclude the paper.

II. RELATED WORKS

Prior works that quantify miscorrelations of SI-induced delay between different analytical timing models or timing tools are limited.

An analytical model that captures SI-induced delay is due to Sapatnekar [10]; it lumps coupling capacitance to ground with the value of Miller coupling factor being 0, 1 or 2 based on the timing window overlap and switching directions of the signals. The effect of crosstalk on net delay is estimated using an iterative algorithm with runtime that is polynomial in the number of nets. The results are not verified with results from other tools or models. Xiao et al. [16] derive an analytical two-pole model for RC interconnect noise waveform calculation with coupling capacitance. A Newton-Raphson iteration is used to obtain the timing information.

Cell / net name	DTran (ns)	SI Incr Delay (ns)	Non-SI Incr Delay (ns)	SI Path Delay (ns)	Non-SI Path Delay (ns)
inst_ram_ctrl_write_ram_fsm_reg_0_/Q	0.000	0.000	0.069	0.269	0.269
inst_ram_ctrl_write_ram_fsm_0_ (net)					
....					
FE_OCP_RBC23542_n28670/Z	0.000	0.000	0.027	0.428	0.428
FE_OCP_RBN23542_n28670 (net)					
FE_OCP_RBC23543_n28670/A	0.004	0.004	0.013	0.445	0.441
....					
U143152/Z	0.000	0.000	0.034	0.809	0.800
n33458 (net)					
U92231/C	0.003	0.002	0.000	0.811	0.801
...					
U99631/Z	0.000	0.000	0.065	0.769	0.762
n33477 (net)					
U145471/C	0.035	0.022	0.002	0.793	0.764
...					
U121581/Z	0.000	0.000	0.104	0.967	0.935
n33452 (net)					
U121579/B	0.133 (0.024)	0.115 (0.021)	0.004	1.082 (0.988)	0.939
U121579/Z	0.000	0.000	0.057	1.139 (1.045)	0.996
n79492 (net)					
inst_ram_ctrl_inst_generic_sp_ram_0_q_reg_21_/D	0.000	0.000	0.000	1.139 (1.045)	0.996

Fig. 4. Timing divergence in a path with the maximum delta slack of 143ps at a clock period of 0.8ns. As defined in Table I, “DTran” is the delta transition due to coupling, “SI Incr Delay” is the incremental delay due to coupling, “Non-SI Incr Delay” is the incremental delay without coupling, “SI Path Delay” is the accumulated path delay with coupling and “Non-SI Path Delay” is the accumulated path delay without coupling. The nets in green color do not contribute to “DTran” and “SI Incr Delay”, whereas the nets in brown color cause non-zero “DTran” and “SI Incr Delay”. The nets that contribute to the delta slack of 143ps are highlighted inside the blue boxes. The values in green are for the same path but analyzed at a clock period of 1.0ns.

Cell / net name	DTran (ns)	SI Incr Delay (ns)	Non-SI Incr Delay (ns)	SI Path Delay (ns)	Non-SI Path Delay (ns)
inst_ram_ctrl_write_ram_fsm_reg_0_/Q	0.000	0.000	0.069	0.269	0.269
inst_ram_ctrl_write_ram_fsm_0_ (net)					
....					
FE_OCP_RBC23542_n28670/Z	0.000	0.000	0.027	0.428	0.428
FE_OCP_RBN23542_n28670 (net)					
FE_OCP_RBC23543_n28670/A	0.004	0.004	0.013	0.445	0.441
....					
U143152/Z	0.000	0.000	0.034	0.809	0.800
n33458 (net)					
U92231/C	0.003	0.002	0.000	0.811	0.801
...					
U99631/Z	0.000	0.000	0.065	0.769	0.762
n33477 (net)					
U145471/C	0.035	0.022	0.002	0.793	0.764
...					
U121581/Z	0.000	0.000	0.104	0.963	0.935
n33452 (net)					
U121579/B	0.024	0.021	0.004	0.988	0.939
U121579/Z	0.000	0.000	0.057	1.045	0.996
n79492 (net)					
inst_ram_ctrl_inst_generic_sp_ram_0_q_reg_21_/D	0.000	0.000	0.000	1.045	0.996

Fig. 5. The path with delta slack of 143ps at clock period of 0.8ns has delta slack of 49ps at clock period of 1.0ns.

Correlation with SPICE shows good matching. However, the Newton-Raphson iteration is computationally expensive and may not be practical for use with realistic designs.

Thiel et al. [12] leverage the ability of PrimeTime (PT) [27] to output a SPICE netlist, and use SPICE simulation to calibrate the PT timing report. However, SI effects are not addressed in this work. Motassadeq et al. [2] extend this analysis flow by using PrimeTime SI (PTSI) [27] instead of PT to include SI effects. Mohamed et al. [9] correlate PTSI-reported delta delay with coupling capacitance and drive strengths of the aggressor and victim. However, they do not provide a quantitative model for these correlations. Venugopal et al. [14] characterize delays calculated by PTSI and correlated with HSPICE [26], but no model predicting the discrepancy of HSPICE and PTSI is presented.

To minimize the gap between internal incremental STA tool and signoff timing tool, Kahng et al. [7] use least-squares regression to model wire delay. They then use offset-based correlation with a signoff timing tool to calibrate the path slacks reported by the internal STA tool. However, they do not explicitly model signoff

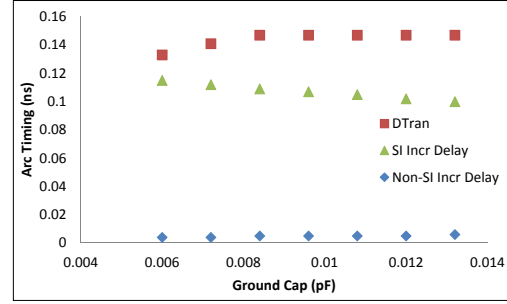
tools in SI mode.

To correlate different signoff timing tools, Mishra et al. [8] recalculate clock uncertainties based on mis-correlation between different tools, and then use the new uncertainty values for better timing correlation between the tools. Han et al. [4] provide a deep learning methodology to correlate timing between different signoff timers. However, they only correlate either non-SI to non-SI mode or SI to SI mode. The models in [4] do not predict timing in SI mode using the timing reports of non-SI mode.

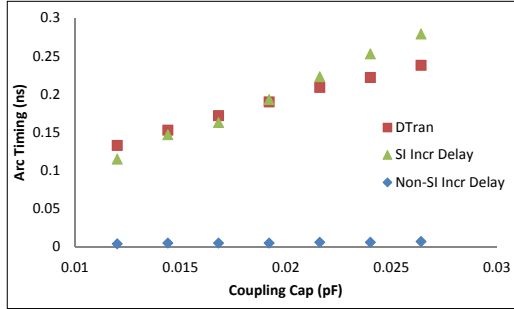
Our work is closely related to that of Han et al. [4], even though the work in [4] does not calibrate non-SI to SI. The key differences are: (i) a new model for incremental transition time due to SI; (ii) a new model for incremental delay due to SI; (iii) a new model for SI-aware path delay; and (iv) validation with a wide range of testcases that include memories from 28nm foundry FDSOI libraries. The new models help us achieve higher modeling accuracy in calibrating non-SI to SI as compared to the models in [4].

Cell / net name	DTran (ns)	SI Incr Delay (ns)	Non-SI Incr Delay (ns)	SI Path Delay (ns)	Non-SI Path Delay (ns)
inst_ram_ctrl_write_ram_ptr_reg_0/Q	0.000	0.000	0.087	0.285	0.285
inst_ram_ctrl_write_ram_ptr_0_(net)					
....					
FE_RC_3395_0/Z	0.000	0.000	0.015	0.424	0.424
FE_OCP_RBN22308_n20174(net)					
FE_OCP_RBN22308_n20174/A	0.014	0.009	0.019	0.452	0.443
....					
U98160/Z	0.000	0.000	0.029	0.541	0.532
n22678_(net)					
FE_OFC16-76_n22678/C	0.003	0.002	0.000	0.543	0.532
...					
U99420/Z	0.000	0.000	0.053	0.742	0.731
n25563_(net)					
U145193/C	0.016	0.012	0.000	0.754	0.731
U145193/Z	0.000	0.000	0.114	0.868	0.845
n25556_(net)					
U89670/B	0.089	0.058	0.006	0.932	0.851
...					
U121246/Z	0.000	0.000	0.021	1.063	0.982
n70246_(net)					
inst_ram_ctrl_inst_generic_sp_ram_1_q_reg_18_/D	0.000	0.000	0.000	1.063	0.982

Fig. 6. Timing divergence in a path with delta slack of 81ps at clock periods of both 1.0ns and 0.8ns.



(a)



(b)

Fig. 7. Timing of the victim net that has the maximum divergence at a clock period of 0.8ns when only (a) ground capacitance and (b) coupling capacitance of the victim net is varied. The figure shows delta transition due to coupling as “DTran” in brown rectangles, arc delay due to coupling as “SI Incr Delay” in green triangles and arc delay without coupling is as “Non-SI Incr Delay” in blue diamonds.

III. METHODOLOGY FOR TIMING CORRELATION IN SI MODE

Our modeling methodology includes (i) selection of parameters that affect incremental delay in SI mode, and (ii) application of nonlinear modeling techniques to capture the complex interactions of parameters so as to accurately predict the incremental delay in SI mode.

A. Selection of parameters

We have studied multiple electrical and circuit parameters that can affect incremental delay in SI mode and have drawn from the list of parameters used to model wire delay in SI mode in [4].

Our analyses indicate that the transition time at the output pin of a net’s driver, the product of wire resistance and capacitances, are not sufficient to predict the incremental delay in SI mode. Figures 8(a) and (b) show that the incremental delay in SI mode vary in the same way for two of the parameters used in [4]. In addition, signoff timing tools use complex algorithms to determine timing windows for less pessimistic delay analyses in SI mode. This is difficult to model because timing windows change with operating conditions. We introduce new electrical parameters to approximate the effect of timing windows for the aggressor with the largest coupling capacitance. Figures 9(a)–(d) show two new electrical and two new structural parameters that affect the incremental delay in SI mode.

We use the following 12 parameters in our modeling: (i) incremental delay in non-SI mode; (ii) transition time in non-SI mode; (iii) clock period; (iv) resistance; (v) coupling capacitance; (vi) ratio of coupling-to-total capacitance; (vii) toggle rate; (viii) number of aggressors; (ix) ratio of the stage in which the arc of the victim net appears to the total number of stages in the path; (x) logical effort of the net’s driver; and (xi), (xii) the differences in max (respectively, min) arrival times¹ of the signal at the driver’s output pin for the victim and its strongest aggressor.² We choose our parameters based on sensitivity of the parameter to incremental transition time or incremental delay due to SI, or SI-aware path delay. Our experimental results indicate that dropping any of the parameters can reduce the modeling accuracy by at least 5%. Therefore, we use all the parameters indicated in Equations (1), (2) and (3) to develop our models. We do not use any layout parameters since layout is reflected in parameters such as coupling capacitance, total capacitance and wire resistance.

We model the incremental transition time due to SI as

$$\Delta T_{si} = f(T_{si}', R_w, C_c, r_{C_c, C_{tot}}, clkp, LE) \quad (1)$$

We further model the incremental delay due to SI as

$$\Delta D_{si} = f(\Delta D_{si}', \Delta T_{si}, R_w, C_c, r_{C_c, C_{tot}}, r_{S, N_{ag}}, clkp, \Delta arr_{min, (r, f)}, \Delta arr_{max, (r, f)}, A_r, LE) \quad (2)$$

¹We use rise and fall arrival times based on the signal’s transition at the output pin of the net’s driver, from timing reports in non-SI mode.

²We consider the net with largest coupling capacitance to the victim as the strongest aggressor.

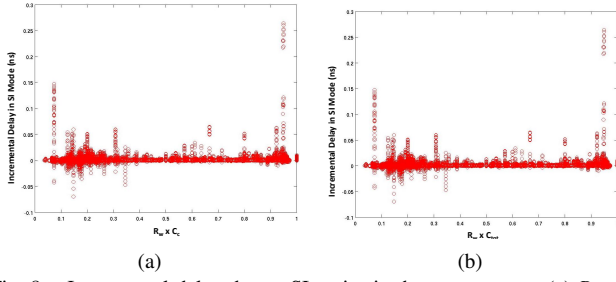


Fig. 8. Incremental delay due to SI varies in the same way as (a) $R_w \times C_c$ and (b) $R_w \times C_{tot}$.

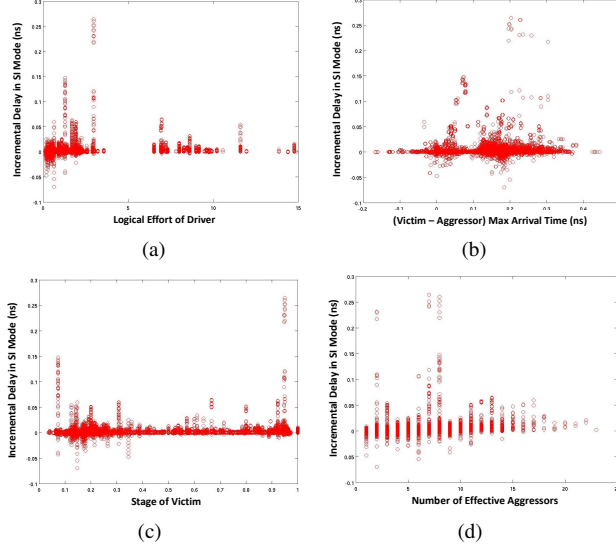


Fig. 9. Incremental delay due to SI varies with (a) logical effort of the net's driver, (b) the difference in max arrival times of victim and the strongest aggressor, (c) the stage in which the arc appears, and (d) the number of effective aggressors of the victim net.

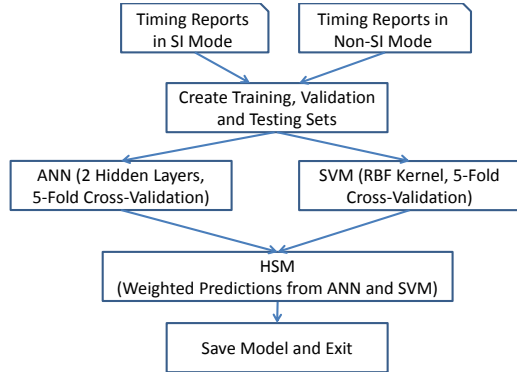


Fig. 10. Modeling flow using nonlinear modeling techniques.

and the SI-aware path delay as

$$\Delta P_{si} = f(P_{si}', \sum_{i=1}^{N_{stg}} \Delta D_{si}, N_{stg}) \quad (3)$$

where ΔD_{si} is the predicted incremental delay due to SI per arc, obtained from the model developed using Equation (2).

B. Nonlinear modeling technique

If the coupling capacitance is zero, we set the incremental delay due to SI as zero; otherwise, we proceed with modeling. We use nonlinear modeling techniques to model the incremental transition time, incremental delay due to SI, and SI-aware path delay, given the complex interactions between modeling parameters described above. For example, reducing the clock period can increase the toggle rates

of both the victim and aggressor nets, and can change the timing windows. As a result, the number of aggressors on the victim can increase. These interactions are non-obvious and cannot be captured by linear modeling techniques. We therefore use Artificial Neural Networks (ANN) and Support Vector Machines (SVM) [5] for our modeling.

In ANN, we use one input, one output and two hidden layers. In each hidden layer, we use up to twice the number of neurons as the number of input parameters. We search for the minimum number of neurons per hidden layer that can achieve the smallest mean-squared error on the training set. In SVM, we use the Radial Basis Function (RBF) kernel with a gamma value of the inverse of the number of the parameters. To generalize our models and avoid overfitting, we use five-fold cross validation and use a separate validation set to reduce overfitting while training our models. We use Hybrid Surrogate Modeling (HSM) [6] to combine the predicted values from the ANN and SVM models and obtain the final predictions. For each technique (ANN, SVM and HSM), we create one model for $N_{stg} \leq 20$ and another model for $N_{stg} > 20$, as our separate studies indicate that modeling accuracy improves with this approach.

IV. EXPERIMENTAL SETUP AND RESULTS

We now describe our design of experiments, i.e., our testcases, methodology to generate “ground truth”, and tool settings. We then describe our modeling results.

A. Design of experiments

In our experiments, we use six real designs (*aes_cipher_top*, *dec_viterbi*, *jpeg_encoder* and *THEIA* from OpenCores [23]; *FIFO* from Synopsys Designware [25]; and single core of *OST2* [24]) as well as artificial testcases developed in-house based on [4]. An illustration of our artificial testcase is shown in Figure 11. We use 28nm FDSOI foundry technology libraries for all our experiments. We vary parasitics, i.e., R_w , C_c , C_g , size of the driver, type of the driver cell, the number of fanouts, clock period, etc. We use default values of 1Ω for R_w , 1fF for C_c and C_g and use scaling factors f_{R_w} , f_{C_c} and f_{C_g} to respectively scale R_w , C_c and C_g in both real designs and artificial testcases.

We use one implementation of the *aes_cipher_top* design signed off at 1.0ns ($\sim 13\text{K}$ standard cells at post-synthesis), one implementation of the *dec_viterbi* design signed off at 1.0ns ($\sim 97\text{K}$ standard cells at post-synthesis), one implementation of the *jpeg_encoder* design signed off at 0.8ns ($\sim 62\text{K}$ standard cells at post-synthesis), one implementation of the *FIFO* design signed off at 0.75ns ($\sim 6.5\text{K}$ standard cells at post-synthesis), one implementation of the *THEIA* design signed off at 2.0ns ($\sim 125\text{K}$ standard cells at post-synthesis) and one implementation of the *OST2* design signed off at 2.2ns ($\sim 350\text{K}$ standard cells at post-synthesis). Table II lists the ranges of various parameters that we sweep in our experiments.

TABLE II
KEY PARAMETERS SWEPT IN OUR EXPERIMENTS.

Parameter	Range	Design/Testcase
$clkp$	$1.0\text{ns} + \{-0.2, 0.2\}\text{ns}$	<i>aes_cipher_top</i>
	$1.0\text{ns} + \{-0.2, -0.1, 0.0, 0.1, 0.2\}\text{ns}$	<i>dec_viterbi</i> , artificial
	$0.8\text{ns} + \{-0.2, -0.1, 0.0, 0.1\}\text{ns}$	<i>jpeg_encoder</i>
	$0.75\text{ns} + \{-0.15, 0.15\}\text{ns}$	<i>FIFO</i>
	$2.0\text{ns} + \{-0.2, 0.2\}\text{ns}$	<i>THEIA</i>
	$2.2\text{ns} + \{-0.2, 0.2\}\text{ns}$	<i>OST2</i>
N_{stg}	$\{15, 20, 25, 30\}$	artificial
$f_{C_c, red}$	$\{0.0, 1.0, 2.0\}$	all
f_{R_w}	$\{0.5, 1.0, 2.0\}$	all
f_{C_c}	$\{0.5, 1.0, 1.5, 2.0\}$	all
f_{C_g}	$\{0.5, 1.0, 2.0\}$	all
Driver size	$\{X6, X16, X24, X32\}$	artificial

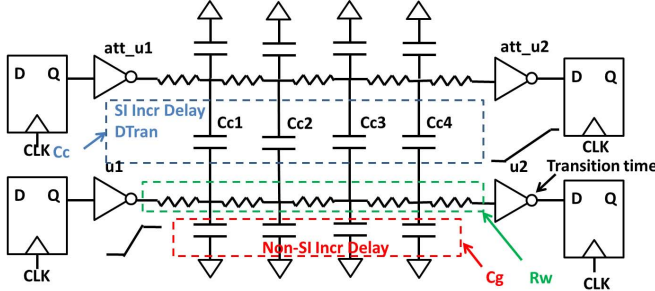


Fig. 11. Illustration of an artificial testcase instance.

To generate “ground truth” data, we perform path-based setup timing analyses in both SI and non-SI modes and report the top-1000 critical paths. In non-SI mode, we use $f_{C_c,red}$ values of 0.0, 1.0 and 2.0 to capture the following effects of victim and aggressor nets switching: (i) a value of 0.0 when the victim and aggressor switch in the same direction; (ii) a value of 1.0 when the victim does not switch but the aggressor switches; and (iii) a value of 2.0 when the victim and aggressor switch in the opposite directions. In SI mode, we use recommended tool settings for the most accurate (least pessimistic) analysis, which include (i) disabling of critical path reselection so that all aggressors are selected for analysis at all times for all victims; (ii) enabling the clock network for analysis so as to include coupling effects of clock nets on victim signal nets; and (iii) performing analysis in edge-alignment mode so as to consider all possible edge arrivals from the upstream logic, using minimum-delay (respectively, maximum-delay) edges for the minimum (respectively, maximum) incremental delay calculations.

Following are steps used for timing analysis in SI and non-SI mode. We specifically highlight the differences in SI versus non-SI mode, if any, in each of the steps.

- *Step 1.* Read databases of timing libraries.
- *Step 2.* Read and link the design; the post-layout netlist is in .v format.
- *Step 3.* Read the constraints specified in .sdc format.
- *Step 4.* Read the parasitics specified in .spef format. In SI mode, read the coupling capacitances, whereas in non-SI mode convert coupling capacitances to ground capacitances by using Miller coupling factor $f_{C_c,red}$.
- *Step 5.* (SI mode only) Set flag to reselect critical paths for SI analysis to false.
- *Step 6.* (SI mode only) Set flag to reselect clock nets for SI analysis to true.
- *Step 7.* (SI mode only) Set flag for delay analysis mode to be edge-aligned.
- *Step 8.* Perform path-based timing analysis of specified top-1000 paths of the signed off design.
- *Step 9.* Report capacitance, incremental delay, transition time, accumulated stage delay of all cells and nets in the top-1000 paths. In SI mode, report incremental delay and transition time due to coupling.

We generate a total of 188K data points of nets that have non-zero value of incremental SI delay, out of which we use 60% for training, 10% for validation and the remaining 30% for testing. The training time of our models is 10.6 hours for ANN, 23.9 hours for SVM and 12 minutes for HSM on an Intel Xeon E5-2640 2.5GHz server with eight threads. This is a one-time overhead. After the models are trained, the time to test is ~ 10 minutes for every 10K data points.

We conduct three experiments to demonstrate accuracy and robustness of our models.

- **Experiment 1. (Accuracy)** Predict incremental transition time, incremental delay and path delay due to SI using a model derived from non-SI timing reports of a signoff timing tool.
- **Experiment 2. (Robustness)** Predict incremental delay due to SI on “unseen” data points from a new implementation of *jpeg_encoder*. The new implementation of *jpeg_encoder* uses different signoff and layout constraints as compared to the implementation (cf. Table II) used to train the models.
- **Experiment 3. (Accuracy)** Compare the predictions of incremental delay and path delay due to SI of our models versus those of [4].

In our results, we compare path delay instead of path slack because the delta in slack arises due to differences in path delay. The required arrival times calculated in both SI and non-SI modes are the same because elements such as clock uncertainty, clock skew, and setup time of the capture flip-flop do not vary with coupling. Only the arrival times vary due to incremental delay in SI and non-SI modes. Therefore, the errors in correlating path slack will be the same as the errors observed in correlating path delay. We report predicted values of transition time and incremental delay due to SI and SI-aware path delay only on the test dataset, that is, we do not include the training and validation datasets in reporting results in Experiments 1 and 2. We calculate percentage error in predicting incremental delay and transition time due to SI in an arc and SI-aware path delay as follows.

$$\text{Error}_{arc} = \frac{(\text{Predicted} - \text{Actual}) \Delta T_{si} \text{ or } \Delta D_{si}}{\text{Actual } \Delta T_{si} \text{ or } \Delta D_{si}} \quad (4)$$

$$\text{Error}_{path} = \frac{(\text{Predicted} - \text{Actual}) \Delta P_{si}}{\text{Actual } \Delta P_{si}} \quad (5)$$

B. Results of Experiment 1

The goal of this experiment is to validate our modeling accuracy in predicting incremental transition time, incremental delay due to SI and SI-aware path delay. Our models are developed by using timing reports in non-SI mode. We test the accuracy of our models by using $\sim 17K$ data points for incremental transition time and incremental delay and ~ 320 paths for SI-aware path delay, across real designs and artificial testcases.

Figure 12 shows actual versus predicted incremental transition times due to SI. Our modeling predictions have a worst-case absolute error of 7.0ps (8.8%)³ and have a range of errors of 11.3ps. Our average absolute error in predicting incremental transition time is 0.7ps (0.6%). Figure 13 shows actual versus predicted incremental delays due to SI. Our modeling predictions have a worst-case absolute error of 5.2ps (15.7%) and have a range of errors of 9.8ps. Our average absolute error in predicting incremental delay is 1.2ps (1.1%).

Figure 14 shows actual versus predicted SI-aware path delays. Our modeling predictions have a worst-case absolute error of 8.2ps (6.9%)⁴, i.e., our worst-case absolute error in predicting path slack is also 8.2ps. The average absolute error in predicting path delay is 1.7ps (1.4%). Figure 15 shows the actual and predicted values of incremental delay and path delay in SI mode of the same path as shown in Figure 4. The path slack divergence between SI and non-SI modes of 143ps is reduced to 5ps by our models.

³In non-SI and SI modes the transition times are 34.6ps and 114.6ps, respectively. The actual incremental transition time due to SI is $114.6 - 34.6 = 80ps$, whereas our model for incremental transition time predicts 73ps. The difference is 7.0ps. Therefore, per Equation (4), the percentage error is $7.0/80 = 8.8\%$.

⁴In non-SI and SI modes the path delays are 1055.2ps and 935.5ps, respectively. The actual difference in SI-aware path delay is $1055.2 - 935.5 = 119.7ps$, whereas our model for SI-aware path delay predicts 109.6ps. The difference is 8.2ps. Therefore, per Equation (5), the percentage error is $8.2/119.7 = 6.9\%$.

Cell / net name	(Actual) SI Incr Delay (ns)	(Model) SI Incr Delay (ns)	(Actual) SI Path Delay (ns)	(Model) SI Path Delay (ns)
inst_ram_ctrl_write_ram_fsm_reg_0_/Q	0.000	0.000	0.269	0.269
inst_ram_ctrl_write_ram_fsm_0_ (net)				
....				
FE_OCP_RBC23542_n28670/Z	0.000	0.000	0.428	0.428
FE_OCP_RBN23542_n28670 (net)				
FE_OCP_RBC23543_n28670/A	0.004	0.004	0.445	0.445
....				
U143152/Z	0.000	0.000	0.809	0.809
n33458 (net)				
U92231/C	0.002	0.002	0.811	0.811
...				
U99631/Z	0.000	0.000	0.769	0.769
n33477 (net)				
U145471/C	0.022	0.023	0.793	0.794
...				
U121581/Z	0.000	0.000	0.967	0.968
n33452 (net)				
U121579/B	0.115	0.118	1.082	1.086
U121579/Z	0.000	0.000	1.139	1.140
n79492 (net)				
inst_ram_ctrl_inst_generic_sp_ram_0_q_reg_21_/D	0.000	0.000	1.139	1.144

Fig. 15. Actual and predicted values of “SI Incr Delay” and “SI Path Delay” (defined in Table I) of the same path shown in Figure 4. Our models reduce the path delay (as well as path slack) divergence from 143ps to 5ps. The predicted values that differ from the actual values are highlighted in red.

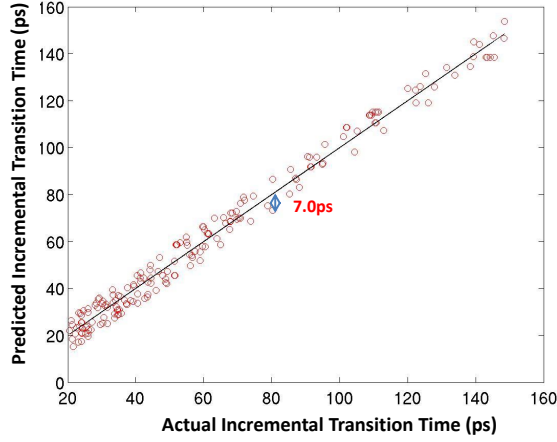


Fig. 12. Actual versus predicted incremental transition times due to SI.

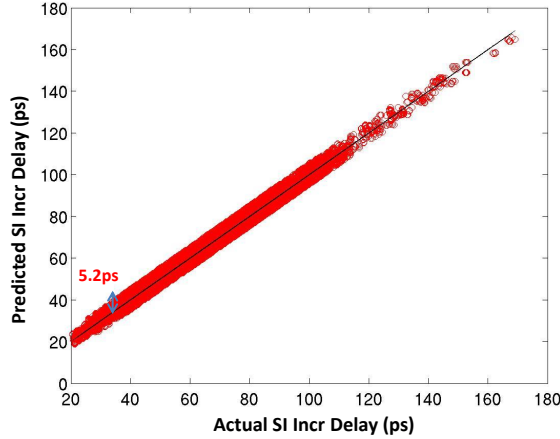


Fig. 13. Actual versus predicted incremental delays due to SI.

C. Results of Experiment 2

The goal of this experiment is to validate the robustness of our models and stress test our models on “unseen” data points. We train our models using data points from our design of experiments described in Section IV-A, and test the models using unseen data points from a new implementation of *jpeg_encoder* signed off with clock period 1.0ns, tighter maximum transition constraint of 150ps

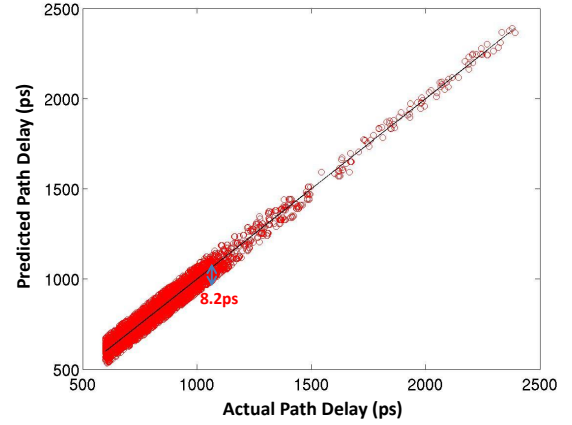


Fig. 14. Actual versus predicted SI-aware path delays.

and utilization of 55%. The implementation used for testing is signed off at different clock period, has different mixes of cell types, number of stages per path, net parasitics, etc. as compared to the implementation (cf. Table II) used to train our models. However, as we include important parameters that affect incremental transition time, incremental delay due to SI, and SI-aware path delay, we expect that our models can be generalized to unseen data points at the same 28nm FDSOI foundry technology. Figure 16(a) shows actual and predicted values of incremental delay in SI mode for 2.5K unseen data points. The worst-case absolute error in prediction is 7.9ps (12.3%), however, the average absolute error is 1.6ps (2.6%). Figure 16(b) shows the distribution of errors across all test data points.

D. Results of Experiment 3

In this experiment we compare the accuracy of our models, versus that of the wire and path delay models in [4] that predict SI-aware path delay. We develop these models for wire and path delay using timing reports in non-SI mode. Recall that Figure 2 in Section I shows that the worst-case error in predicting incremental arc delay due to SI using the model in [4] can be as large as 60ps. Figure 17 shows that the worst-case error in path delay can be 87.3ps using the model in [4]. As described in Section IV-B, our models have worst-case errors of 5.2ps and 8.2ps in predicting incremental delay

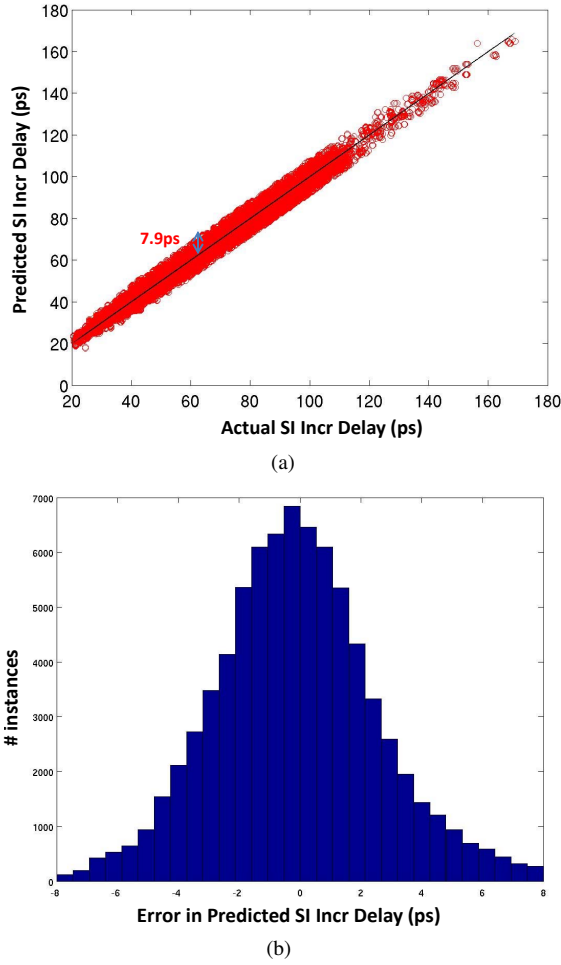


Fig. 16. Robustness of our models in predicting incremental delays due to SI. (a) Actual versus predicted, and (b) distribution of modeling errors.

due to SI, and SI-aware path delay, respectively.

The models of [4] have large prediction errors in spite of using a layered modeling approach. We attribute this to underfitting, with the parameters used in [4] being insufficient to capture fully the variations in incremental delay due to SI, and SI-aware path delay.

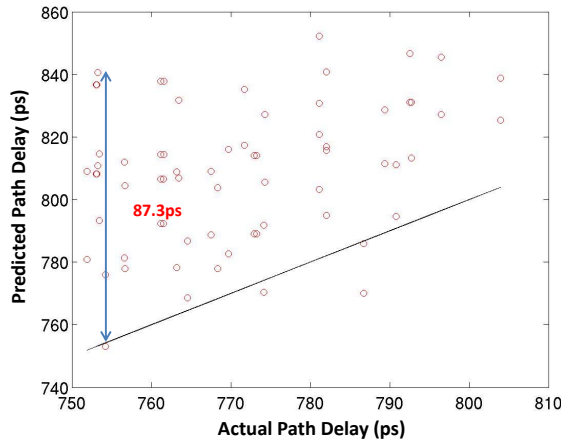


Fig. 17. Actual SI-aware path delays, versus predicted path delays using models of [4].

V. CONCLUSIONS

In this work, we analyze electrical and logic structure parameters that cause timing in non-SI mode to diverge from that in SI

mode. We provide a machine learning-based methodology that can accurately model incremental delay due to SI, and SI-aware path delay. Our models for a 28FDSOI production technology and cell library have worst-case errors of 7.0ps, 5.2ps and 8.2ps, respectively in predicting incremental transition time, incremental delay due to SI, and SI-aware path delay. We demonstrate that our models are more accurate than previous work [4]. Our ongoing works include (i) predicting timing reports in path-based analysis using reports from graph-based analysis, and (ii) integrating our models with an academic timer [28].

REFERENCES

- [1] M. Becer, V. Zolotov, R. Panda, A. Grinshpon, I. Algor, R. Levy and C. Oh, "Pessimism Reduction in Crosstalk Noise Aware STA", *Proc. ICCAD*, 2005, pp. 954-961.
- [2] T. El Motassadeq, V. Sarathi, S. Thameem and M. Nijam, "SPICE versus STA Tools: Challenges and Tips for Better Correlation", *Proc. SOC Conf.*, 2009, pp. 325-328.
- [3] R. Gandikota, K. Chopra, D. Blaauw and D. Sylvester, "Victim Alignment in Crosstalk-Aware Timing Analysis", *IEEE Trans. CAD* 29(2) (2010), pp. 261-274.
- [4] S. S. Han, A. B. Kahng, S. Nath and A. Vydyanathan, "A Deep Learning Methodology to Proliferate Golden Signoff Timing", *Proc. DATE*, 2014, pp. 1-6.
- [5] T. Hastie, R. Tibshirani and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, New York, Springer, 2009.
- [6] A. B. Kahng, B. Lin and S. Nath, "Enhanced Metamodeling Techniques for High-Dimensional IC Design Estimation Problems", *Proc. DATE*, 2013, pp. 1861-1866.
- [7] A. B. Kahng, S. Kang, H. Lee, S. Nath, J. Wadhvani, "Learning-Based Approximation of Interconnect Delay and Slew in Signoff Timing Tools", *Proc. SLIP*, 2013, pp. 1-8.
- [8] A. Mishra, J. Kumar and U. Singhal, *Resolving Timing Mismatch Using Timing Uncertainties*. <http://www.edn.com/design/integratedcircuit-design/4390721/Resolving-timing-mismatch-using-timinguncertainties>
- [9] S. A. Mohamed, A. A. Manaf and C. C. Teh, "A Noise and Signal Integrity Verification Flow for Hierarchical Design", *Proc. ICCAIE*, 2011, pp. 250-255.
- [10] S. S. Sapatnekar, "Capturing the Effect of Crosstalk on Delay", *Proc. VLSI*, 2000, pp. 364-369.
- [11] G. Smith, Gary Smith EDA, *personal communication*, October 2014.
- [12] T. Thiel, "Have I Really Met Timing-Validating PrimeTime Timing Reports with Spice", *Proc. DATE*, 2004, pp. 114-119.
- [13] K. Tseng and M. Horowitz, "False Coupling Exploration in Timing Analysis", *IEEE Trans. CAD* 24(11) (2005), pp. 1795-1805.
- [14] C. R. Venugopal, P. Soraiyur and J. Rao, "Evaluation of the PTSI Crosstalk Noise Analysis Tool and Development of an Automated Spice Correlation Suite to Enable Accuracy Validation", *Proc. ISQED*, 2008, pp. 334-337.
- [15] T. Xiao and M. Marek-Sadowska, "Worst Delay Estimation in Crosstalk Aware Static Timing Analysis", *Proc. ICCD*, 2000, pp. 115-120.
- [16] T. Xiao and M. Marek-Sadowska, "Efficient Delay Calculation in Presence of Crosstalk", *Proc. ISQED*, 2000, pp. 491-497.
- [17] *Atrenta Inc.* <http://www.atrenta.com>
- [18] *Cadence Design Systems.* <http://www.cadence.com>
- [19] *CLK Design Automation Inc.* <http://www.clkda.com>
- [20] *Incentia Design Systems Inc.* <http://www.incentia.com>
- [21] *Matlab.* <http://www.mathworks.com>
- [22] *Mentor Graphics Inc.*, <http://www.mentor.com>
- [23] *OpenCores.* <http://opencores.org>
- [24] *OpenSPARC T2* <http://www.oracle.com/technetwork/systems/opensparc/index.html>
- [25] *Synopsys.* <http://www.synopsys.com>
- [26] *Synopsys HSPICE User Guide.* <http://www.synopsys.com/Tools/Verification/AMSVVerification/CircuitSimulation/HSPICE/Pages/default.aspx>
- [27] *Synopsys PrimeTime User Guide.* <http://www.synopsys.com/Tools/Implementation/SignOff/Pages/PrimeTime.aspx>
- [28] *UCSD Gate Sizer*, <http://vlsicad.ucsd.edu/SIZING>