Silicon Speedpath Measurement and Feedback into EDA flows

Kip Killpack, Chandramouli Kashyap, Eli Chiprout
Intel Strategic CAD Labs, Hillsboro, OR
kip.killpack@intel.com

ABSTRACT

Timing, test, reliability, and noise are modeled and abstracted in our design and verification flows. Specific EDA algorithms are then designed to work with these abstracted models, often in isolation of other effects. However, tighter design margins and higher reliability issues have increased the need for accurate models and algorithms. We propose utilizing silicon data to tune and improve the EDA tools and flows. In this paper we describe a silicon methodology to isolate silicon speedpath environments and feed these into a simulation framework to temporally and spatially isolate specific speedpaths in order to model and understand the real effects. This is done using accurate electrical speedpath modeling techniques which may be used to tune the accuracy and correlation of the design models. The effort required to distinguish the many different electrical effects will be outlined.

Categories & Subject Descriptors:

B.7.2 Design Aids – Simulation, Verification, B.8.1 Reliability, Testing and Fault-Tolerance, B.8.2 Performance Analysis and Design Aids

General Terms:

Performance, Verification, Design, Measurement.

Keywords:

Silicon, Speedpath, Timing, Correlation, Measurement.

1. INTRODUCTION

Abstraction has played a central role in the modeling of electrical effects. The typical approach has been to start with fundamental physical equations and to abstract the low level models into something that an algorithm operating at a higher level of abstraction can use efficiently. For example, on the extraction front, Maxwell's equations are often abstracted into resistors, capacitors and inductors for simplifying the modeling process. For timing purposes, interconnect resistors and capacitors may be abstracted into a simple load capacitor or a pi-model[1]. Coupling effects may be abstracted by using Miller coefficients on the coupling capacitors [2]. Gate models that start from the fundamental equation of device physics are abstracted into BSIM models, piecewise linear models, or simple switches [3]. This abstraction process continues all the way up the design hierarchy for early as well as late design. Efficient algorithms are then designed to analyze the design using these abstracted models. While this paper will concentrate on timing, the same is true for thermal, noise, reliability and other effects.

Additionally, for tractable algorithmic solutions (in terms of CPU time and memory), complex effects that may interact with each other are often modeled in isolation. For example, when analyzing the cross-coupling effects on static timing paths, multiple input switching effects, power delivery effects, thermal effects, etc. are often not considered; effectively ignoring their correlation to cross-coupling effects [4][5][6][7][8]. This is in addition to the point that static timing, by itself, is an abstraction of dynamic electrical effects. Given the reality of requiring tractable solutions, we have come to expect that our models will have a certain level of bounded inaccuracy (where the error bound is known) and even uncertainty (where the error bound is not known). However, two developments have caused this state of affairs to become problematic in the last few years.

First, the number of physical effects impacting timing paths has increased dramatically: capacitive and inductive noise, resistive and dielectric effects of new materials, power grid noise, thermal effects, leakage effects, multiple input switching, random dopant fluctuations, correlated layout effects, process reliability effects, etc. These effects often occur simultaneously and may be highly correlated. Isolating them and abstracting, as we do in our timing flows, may cause considerable inaccuracy. Second, design margins have been reducing in order to get more out of diminishing silicon technology returns. The push toward lower power and higher performance has required that we no longer allow large pessimistic guard bands that have traditionally hidden these inaccuracies.

For this purpose, we propose to use microprocessor silicon in a highly structured way in order to determine the importance of various physical effects for the application under consideration – namely timing for this paper. Given the importance of each effect, we then propose to delve deeper and compare our models to each of the silicon effects in order to enable more accurate correlation between our models /algorithms and the actual measured effects.

Several attempts at correlating models have been made in the industry. However, these usually involve isolated structures that enable singular measurements. While these are useful, the real design impact comes about when the design, implemented on manufactured silicon with all of its variational deficiencies, runs microprocessor instructions that produce multiple unanticipated dynamic and environmental effects. Therefore, we propose in this paper, to isolate actual design speedpaths in silicon and determine their causal effects by an accurate modeling scheme. This will then give us the necessary insight to correct our static and dynamic timing models in order to reduce the design pessimism.

The first part of this paper will give a rough overview of the necessary mechanisms needed to bring in silicon speedpath data to a dynamic timing infrastructure. Then we will describe how we isolate an ambiguous set of potential candidate paths using timing.

DAC 2007, June 4–8, 2007, San Diego, California, USA.

Thirdly, we will describe the algorithmic approach which we use to determine the main causal effect(s) of the speedpath. Finally, we present initial feedback results to the EDA flows and conclude with potential directions and open research for this area.

2. Path isolation

A timing path consists of a sequence of transistors, from a source sequential element to a sink, for which a total delay may be calculated using a timing tool. Millions of such paths are theoretically possible but only some of these can be or are logically exercised. Even fewer are frequency limiting paths. The ideal situation would be for us to observe a logically sensitizable path in operation during a single cycle on the die and determine its individual transistor delays and slopes as well as environmental conditions such as side input switching patterns and timings, coupling neighbors and timings, power grid noise, etc. Using many such observations, we can determine how well our timing tools are doing in predicting the path delay in silicon across multiple cycles. However, in order to observe a path it is neither simple nor desirable to probe it directly. Non-invasive techniques such as laser probing exist but are expensive and don't have a single-cycle resolution. The only economical probe elements that are accessible are digital and are the external pins/bus of the chip as well as any designed, usually sparsely populated, internal scan latches that indicate either correctly captured logic values or a failure somewhere. It is therefore a long and complex deduction path from this information to knowing something about a specific path.

A modern microprocessor die has a quarter billion transistors. Even simple test patterns have millions of cycles (or logic vectors). For a given test pattern, there is a single logically sensitized path of transistors that runs slower than all other paths. The goal of speedpath isolation is to identify the slowest path and the cycle when it is slowest. This section gives an overview of the problem of isolating speedpaths in silicon but given the purpose of this paper to address correlation to timing models, we choose not to describe in detail all the necessary mechanisms for doing so. We leave this to a full paper with our acknowledged partners.

2.1 Isolation on the tester

The tester is the beginning step of the process of isolating speedpaths. On the tester, a test pattern is run on a single die repeatedly. The voltage is fixed and, for each run, the frequency of the part is increased slightly until the part produces an incorrect result on the bus pins. That gives the frequency of failure for the test but not the clock cycle on which it failed because the offending path might have failed several cycles before the incorrect result appeared on the observable pins. The clock cycle number is necessary in order to have a temporal isolation of the speedpath. In order to obtain the clock cycle, the part is run at a frequency just above the failing frequency. Subsequently, on a cycle-by-cycle basis, the clock is shrunk. If a critical speedpath is sensitized on the shrunk cycle, it is forced to fail as shown in Figure 1. This failure propagates and is eventually detected on the external bus as a mismatch between observed and expected data.

Once the failing cycle is known by a shrink and subsequent failure on the bus, scan latch data is extracted. The failing scan latches are identified by their incorrect state values following the shrink cycle. This helps to isolate the speedpath to a region of potential speedpaths on the die but is not complete in that the region may be large. Due to the rarity of scan latches, the non-captured data usually occurs on a non-scan sequential element (latch or flip-flop) and it may take one or more cycles for the faulty latched data to propagate to a scan latch. Thus, one is required to perform further analysis to spatially isolate the speedpath (see Figure 2). Part of this isolation is to determine the logically sensitizable paths that can cause the same failure as observed on the identified scan latch(es) on the tester. This process will not be described in detail in this paper. For the purposes of this paper it is necessary to know that any logical analysis is not sufficient to isolate a path in silicon since, without timing information, there may be two or more potential logical paths that cause the same observed failure on the scan latch, particularly if a path is far from the scan latch. Moreover, logical analysis cannot give insight into the electrical effects and causal pathways due to those effects that impact the delay number observed and which we are interested in for the purpose of this paper.

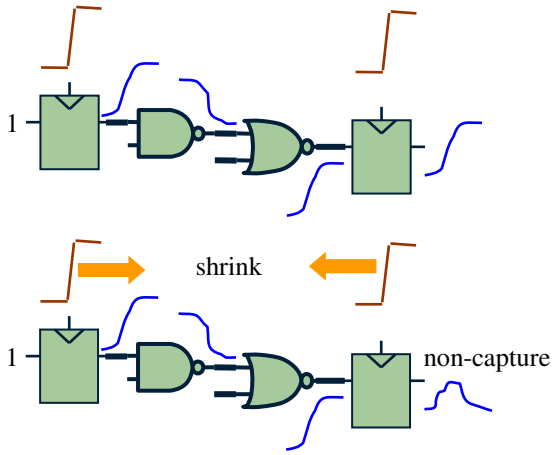

Figure 1. Clock shrink causing non-capture

2.2 Isolation using timing

The logic analysis mentioned in Section 2.1 identifies a set of potential sink elements where the critical speedpath terminates. The cardinality of the set is one or more. For the purpose of final speedpath isolation and in order to get the data of interest to feed back to the timing flows, including relevant electrical causal effects, it is necessary to load the logic and tester information in a "timing infrastructure" that is significantly different than the timing tools used for design. Our internal timing infrastructure called ATARE (Advanced Timing Analysis Research Engines) has been enhanced to meet these new requirements.

Using ATARE, we analyze the cone of logic feeding a potential sequential sink element. Figure 2 depicts one such cone. We perform dynamic simulation of the cone using the logic vectors from the test pattern during the shrink cycle. This is in contrast to a Static Timing Analysis (STA) framework in which worst-case logic sensitization is assumed on each gate in order to bound behavior across all possible logic patterns. Secondly we perform a simulation-based clock shrink to identify the failing frequency of the logic cone. This is identified when the sample latch fails to capture the correct value (Fig. 1). This is in contrast to STA setup checks on sequentials which are defined as a Clk-to-O delay pushout rather than a true failure to capture.

As we are attempting to model a silicon speedpath, we must account for all known effects that can impact timing. We employ a bounding strategy to handle the modeled effects. Through successive improvements to the model, at the cost of more expensive runtime analysis, we are able to shrink the timing bounds. For example we implemented an accurate noise-based coupling model that incorporates logic filtering based on the test pattern. Even without analyzing the arrival time of switching aggressors we can filter out much of the coupling impact and reduce timing bounds. In the situation when logic analysis identifies more than one potential sink latch, the timing bounds are used to further refine the list. In most cases there is a slow cone of logic and the others are non-critical with their upper bound being less than the lower bound of the slowest cone. Thus timing analysis filters the potential sink list to a single sequential element where the speedpath ends.

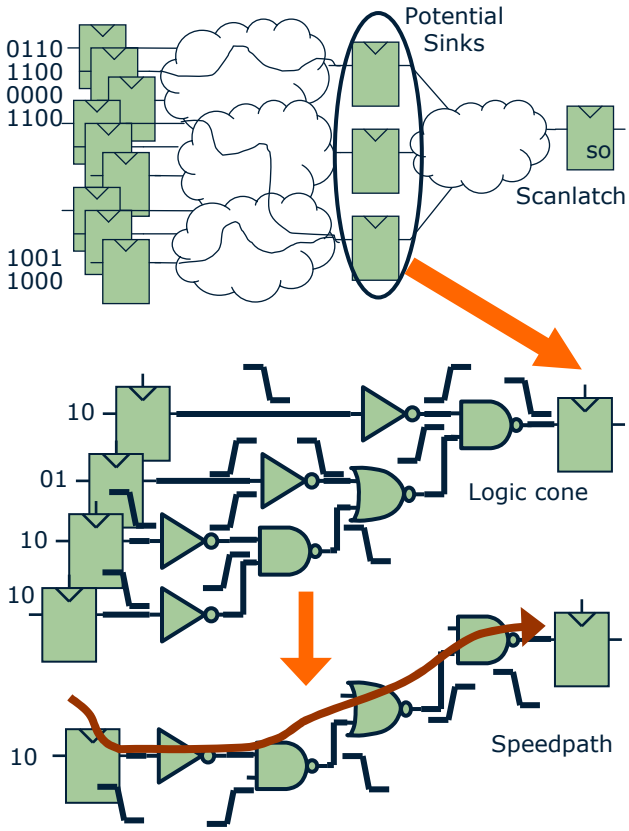

Figure 2. Isolating a speedpath spatially

Once the sink latch is identified, the next step is to identify which gates in the cone cause the failure and which generating sequentials start the path. Logic alone cannot answer this question as there can be many potentially causal transitions in the cone. However, there are usually only a few transitions that have an impact on cone timing and a single speedpath is identified as shown in Figure 2. A detailed description of causality isolation is discussed in Section 3.

In cases where two potential sink sequentials have cones with very similar timing, more detailed timing analysis must be performed to further reduce timing bounds or shift the means. There may be situations in which the most accurate analysis has been performed and there are still multiple cones with overlapping timing bounds. The remaining spread can be due to random effect uncertainties, such as Vt variation, for which an exact value cannot be known or measured. Under this condition it is impossible to isolate the exact speedpath without additional silicon information to help tune the model.

3. Root causing using sensitivities

Consider the logic cone shown in Figure 2. The delay of every gate in the cone is a function of various parameters like Le, Vt, W, Vcc, Temperature, etc. As a result, the arrival time at the output node, $A_{o,}$ is a function of every parameter affecting every gate in the cone along with the logic vector applied to the cone inputs. In this section we derive a linear analytical model that allows us to rank the effects and the gates in the cone that have the most impact on A_o.

To address the linearity assumption on a logic cone, Figure 3 shows the output arrival time of a two input NAND gate as a function of the two input arrival times. The arrival times have been normalized to FO4 delays. Two orthogonal planes can be seen corresponding to when one or the other arrival time dominates implying a linear behavior under this scenario. As expected, the output arrival time shows a non-linear behavior when the input arrival times are close together. However, for *small* changes in the input arrival time, the response surface looks almost linear as the zoomed in graph in Figure 4 shows. We find that most of the time the separation between the arrival times is large enough that one arrival time clearly dominates the other and the linearity assumption holds. When the two arrival times are indeed close enough, the variations in arrival times (due to the parameters described) for the individual measured cycle are small enough that over this range the linearity assumption still holds as seen in Figure 4. As we show later, experimental results support this assumption.

First some remarks on notation. We use bold font to denote both vectors and matrices, with lowercase letters denoting vectors and uppercase denoting matrices. Vectors are always assumed to be column vectors unless noted otherwise. Non-bold letters denote scalars in the equations. Suppose that there are *n* gates in the cone and that each gate depends on *m* parameters. For simplicity, we assume that all the parameters are independent. In reality, for correlated parameters Principle Component Analysis can be performed to get a set of uncorrelated parameters. Let p_{ij} be the j^{th} parameter of the i^{th} gate where $1 \le i \le n$ and $1 \le j \le m$. We assume the parameters are normalized such that $p_{ij} = (p'_{ij} - \ _{ij})/\sigma_{ij}$ where p'_{ij} is the original parameter, $_{ij}$ is the mean of the original parameter and σ_{ij} is the sigma. Assuming small variations of the parameter values around the mean, we can fit a linear model given by:

$$A_o = k_0 + \mathbf{c}^T\mathbf{p} \qquad (1)$$

where **c** and **p** are $mn \times 1$ column vectors. The vector **c** denotes the *sensitivity* of A_o with respect to the parameters affecting the gates in the cone. Once such a model is available, the magnitude of sensitivities can be used to rank the importance of every parameter in the cone that affects A_o. The linearity assumption holds since

gate delays are fairly linear in Le, Vt, Temperature, etc. even for a wide range of variations. Although gate delays vary quadratically with Vcc, for small changes in Vcc linearity is a valid assumption. To test the validity of the above equation, we extracted many cones from a 65nm microprocessor and for each one, performed Monte-Carlo sampling on the Le, Vt, and W parameters of the transistors with one input vector pattern per cone. Using the output response we fit a model of the form shown in **(1)** using a least squares method. A new set of Monte-Carlo samples is then used to test the accuracy of the model versus Spice simulation. The results for one cone are shown in Figure 5, with cone arrival time normalized to FO4 delays. The errors are less than +/- 1%, and similar errors exist on the other tested cones. Clearly, **(1)** provides a very good model for A_o as a function of the device parameters.

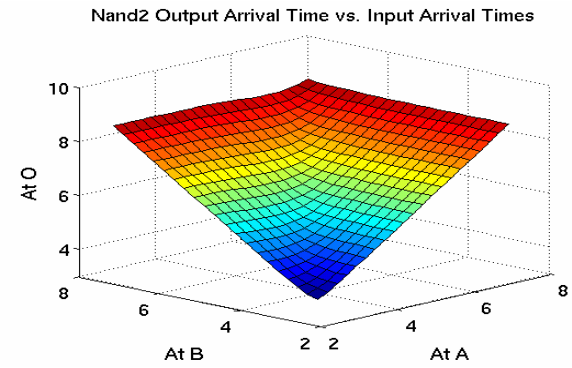

Figure 3. Nonlinear Output Behavior

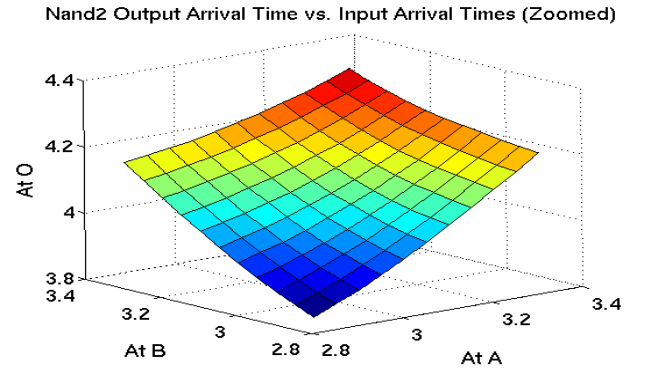

Figure 4. Zoomed in Nonlinear Output Behavior

A drawback of the above approach is the large simulation time required to collect the data to fit **(1)**. A closer look shows that the arrival time A_i at the output of a gate i depends on only a few parameters *local* to the gate. Specifically, A_i depends on the arrival times at the inputs to the gate, the input slews, the parameters of the gate itself like W, Le, Vt, Vcc, etc. as well as the Le, Vt, and W of the gates in its immediate fanout as they affect the load seen by gate i. Since the number of parameters that an individual gate depends on is small, only a few simulations are needed to fit A_i to a linear model. Similarly, we also fit the output slew of the gate to a linear model. Let **e** denote a vector of timing events, where the event could be an arrival time or the slew at the output of a gate. We then have:

$$e_i = k_i + \mathbf{b_i}^T \mathbf{e_i'} + \mathbf{c_i}^T \mathbf{p_i} \qquad (2)$$

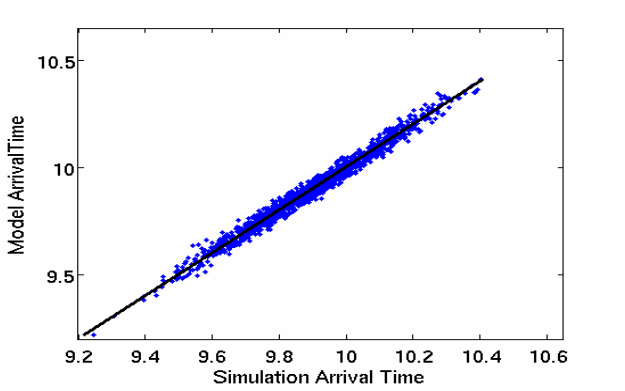

Figure 5. Global Fit Correlation

where $\mathbf{e_i'}$ is the vector of arrival times and slews at the inputs to gate i and $\mathbf{p_i}$ is the set of local parameters on which gate i depends. Given **(2)**, it remains to be shown how we can compute A_o efficiently. With k PIs on the cone, let **i** be a 2k x 1 vector of arrival times and slews at the PIs of the cone. Let **e** be the 2n x 1 vector of arrival times and slews at all the nodes of the circuit, and **p** be the mn x 1 vector of parameters of the circuit. Then the 2n equations of the form **(2)** can be written as a system of equations in the following form:

$$\mathbf{Be + Di + Cp + k = 0} \qquad (3)$$

where **B** is 2n x 2n, **D** is 2n x 2k, **C** is 2n x mn and **k** is 2n x 1. Note that **C** is the sensitivity matrix of the circuit with respect to parameters affecting gates in the cone, and **D** is the sensitivity matrix with respect to the PI arrival times and slews. **(3)** can be solved for the arrival time and slew at every node of the circuit including the output node:

$$\mathbf{e} = -\mathbf{B}^{-1}(\mathbf{Di} + \mathbf{Cp} + \mathbf{k}) \qquad (4)$$

A_o can be computed by picking off the appropriate index of the output node from the vector **e** which symbolically can be written as:

$$A_o = \mathbf{f}^T \mathbf{e} \qquad (5)$$

where **f** is a 2n x 1 vector with a 1 corresponding to the index of A_o and 0s elsewhere.

Figure 6 shows the errors when A_o is computed using **(5)** rather than **(1)**. The figure clearly shows that **(5)** is a good model for computing the arrival time given the arrival times and slews at the PIs and the sensitivities. The errors are less than +/- 1.5%. The combination of local fitting combined with **(5)** gives an efficient procedure for computing A_o. On average, the local fitting method requires an order of magnitude fewer simulations compared to the global fit method. The vector $-\mathbf{f}^T\mathbf{B}^{-1}\mathbf{C}$ gives the mn sensitivities of the gate parameters on which A_o depends. Typically, most of the entries of this sensitivity vector are 0. It is only the entries corresponding to the gates that impact A_o that are non-zero. Most often this corresponds to the gates on a single critical path. However, this method also shows MIS situations when more than one path affects A_o. We discuss this further in the results section. Additionally, the relative magnitude of the sensitivities identifies the causal effects. For example, if the sensitivity to Vcc is larger than all the device Le sensitivities then Vcc has more impact on A_o than Le variation.

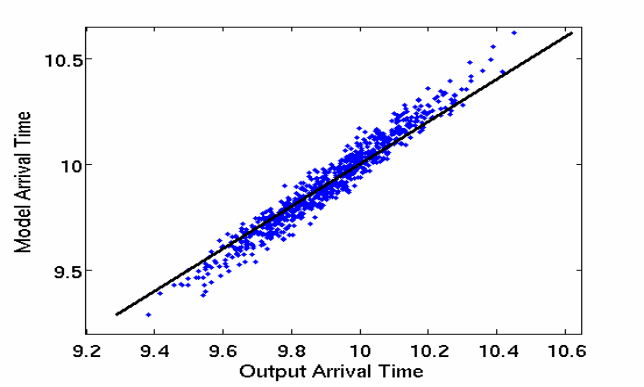

Figure 6. Local Fit Correlation

4. Initial results

Using the speedpath isolation framework described in section 2, we have extracted silicon speedpath measurements from a 65nm Intel microprocessor. Many clock shrinks were performed and a number of these were isolated to specific speedpaths. With these in hand, we created an analytical model and proceeded to extract causality information. One important silicon to timing feedback information that we wanted to obtain was whether multiple input switching (MIS) occurs on these speedpaths. Logically there are many gates in the cone that have more than one input switching. However, due to early arrival times very few MIS situations actually have impact on the cone output arrival time. On 13 out of 16 cones analyzed, the causal path only has a chain of single input switching (SIS) events that affect output arrival time. The other 3 cones have a single gate with MIS events that influence cone arrival time. In these 3 cones, the MIS gate has a dominant path with large coefficients and a sub-path with smaller coefficients as seen in Figure 7. The sub-path has smaller impact on overall timing of the cone, but can be taken into account for design fixes.

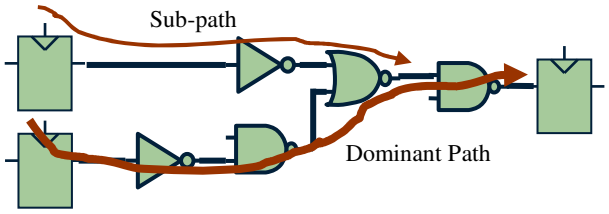

Figure 7. A Multiple Input Switching Path

In our detailed dynamic cone simulation framework we saw that much of the coupling on nets in the cone go to nets outside the cone. An attempt to analyze the cones feeding the aggressors as well as the cone under test leads to an explosion in runtime. Rather we took the approach of identifying an upper bound on coupling noise impact from aggressors outside the cone. We used a noise-based model with conservative assumptions on victim and aggressor slopes and alignment to compute a max pushout for each aggressor. In addition we applied logic filtering from the test pattern on the cycle when the cone was observed in silicon. We evaluated the stages in 4 cones of logic on the cycle when they were critical in silicon and compute a max pushout using the noise model, and also the max pushout using noise and logic filtering.

The two CDFs of the max pushout over all stages are shown in Figure 8. The median pushout when ignoring logic is 4.5% while including logic reduces the median to 0.7%. This shows the effectiveness of applying logic in reducing coupling noise pessimism during vectored cone simulation. Note that even with conservative assumptions on aggressor slope and alignment, using logic and noise analysis shows that 90% of the stages experience a pushout of less than 5%. On stages with large upper bounds, a more expensive and detailed analysis can help further reduce the pushout upper bound.

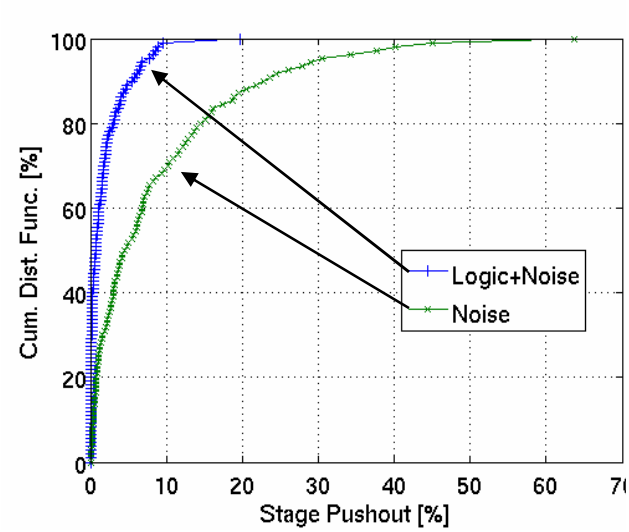

Figure 8. Coupling Pushout

5. Conclusions and future research

We have shown a new framework for analyzing real silicon speedpaths in order to obtain data to feed back into our timing flows. This data is the reality behind the abstracted and isolated models we often use in EDA. This paper outlined in brief a silicon speedpath isolation technique and causality analysis framework in a new kind of silicon-based dynamic timing framework (ATARE). The initial results from this framework are only the tip of the iceberg of insights that may be obtained from silicon and fed back to the EDA flows. We outlined two in this paper: MIS and cross-coupling effects. This approach opens up a rich set of potential research topics: how to influence this framework with process variation data (estimated or based on measured results), how to mathematically narrow the silicon-to-model correlation results, how to use information from multiple paths to determine the best static timing flow, how to use information from multiple cycles on the same path, how to improve bin splits of good die using a combination of silicon debug and silicon-based timing analysis, how to use data from burnt-in (or stressed) silicon parts to better gauge the impact of aging on timing, etc. We would like to encourage the research community to think about this problem and expand the approaches that will finally answer the fundamental question: how good are our flows in really predicting silicon?

6. Acknowledgments

We would like to acknowledge the partnership of Chirayu Amin, Praveen Parvathala, Arun Krishnamachary, and Suriyaprakash Natarajan at Intel.

7. REFERENCES

[1] Dartu, F.; Menezes, N.; Qian, J.; Pillage, L.T, "A Gate-Delay Model for High-Speed CMOS Circuits", Design Automation, 1994. 31st Conference on, 6-10 June 1994 Page(s): 576-580.

[2] Kahng, A.; Muddu, S.; Sarto, E.; "On switch factor based analysis of coupled RC interconnects", Design Automation, 2000. 37th Conference on. Pages: 79-84

[3] Gowda, S.M.; Sheu, B.J, "BSIM plus: an advanced SPICE model for submicron MOS VLSI circuits", Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, Volume 13, Issue 9, Sept. 1994 Page(s):1166 – 1170.

[4] Kouroussis, D.; Ahmadi, R.; Najm, F.N., "Voltage-Aware Static Timing Analysis", Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, Volume 25, Issue 10, Oct. 2006 Page(s):2156 – 2169.

[5] Blaauw, D.; Zolotov, V.; Sundareswaran, S., "Slope propagation in static timing analysis", Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, Volume 21, Issue 10, Oct. 2002 Page(s):1180 – 1195.

[6] Jyu, H.-F.; Malik, S.; Devadas, S.; Keutzer, K.W., "Statistical timing analysis of combinational logic circuits", Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, Volume 1, Issue 2, June 1993 Page(s):126 – 137.

[7] Harris, D.; Horowitz, M.; Liu, D**, "**Timing analysis including clock skew", Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, Volume 18, Issue 11, Nov. 1999 Page(s):1608 – 1618.

[8] Hai Zhou, "Timing analysis with crosstalk is a fixpoint on a complete lattice", Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, Volume 22, Issue 9, Sept. 2003 Page(s):1261 – 1269.