Efficient Generation of Delay Change Curves for Noise-Aware Static Timing Analysis

Kanak Agarwal, Yu Cao[1], Takashi Sato[2], Dennis Sylvester, and Chenming Hu[1]
University of Michigan, Ann Arbor [1]*University of California, Berkeley* [2]*Hitachi, Ltd.*
agarwalk@engin.umich.edu

Abstract

In this paper, we explore the concept of using analytical models to efficiently generate delay change curves (DCCs) that can then be used to characterize the impact of noise on any victim/aggressor configuration. Such an approach captures important noise considerations such as the possibility of delay change even when the switching windows of neighboring gates do not overlap. The technique is model-independent, which we demonstrate by using several crosstalk noise models to obtain results. Furthermore, we extend an existing noise model to more accurately handle multiple aggressors in the timing analysis framework. DCC results from the analytical approach closely match those from time-consuming SPICE simulations, making timing analysis using DCCs efficient as well as accurate.

1. INTRODUCTION

Interconnect noise caused by coupling capacitance can be separated into two forms: 1) *crosstalk*, which we define as a voltage glitch on a quiet victim line, and 2) *dynamic delay*, which refers to the uncertainty in delay of a stage (gate + wire) due to the switching activity of nearby gates. For static CMOS designs, the potential timing errors caused by dynamic delay are as significant as functional implications of crosstalk. Dynamic delay can easily exceed 20-30% of the nominal delay for relatively short wires (< 0.5 mm), depending on driver and interconnect configurations. This degree of delay uncertainty is intolerable for designs with tight timing budgets.

To first order, dynamic delay is proportional to the ratio of coupling capacitance (C_c) to total stage capacitance (including junction, fan-out, and interconnect ground capacitances). The portion of interconnect capacitance attributable to coupling has risen to about 80% in scaled technologies for minimum-pitch wiring, both global and local. Assuming that interconnect capacitance dominates gate loading in global nets, the amount of dynamic delay can reach ±80% of the nominal delay. Figure 1 shows the simulated increase in delay uncertainty for a 3 mm global wire through a number of technology generations. A large inverter with fan-out of 1 serves as both victim and aggressor. Worst-case dynamic delay approaches the 80% plateau, corresponding to the portion of capacitance due to coupling. Since there is only a single aggressor in these simulations, the delay only fluctuates approximately 80/2, or 40%, above or below the nominal delay value.[1]

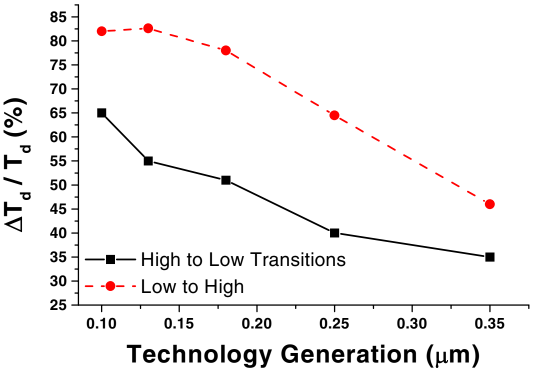

Figure 1. Technology scaling leads to increased dynamic delay effects. Line length is 3 mm.

There are two primary modeling approaches to dynamic delay. The first is based on the Miller effect, which replaces a capacitance between two nodes by equivalent capacitances to ground from each node. In an on-chip context, the coupling capacitance between two adjacent wires is replaced by a ground capacitance for each net. The resulting ground capacitance is traditionally set to either 0 or 2*C_c which were long considered lower and upper bounds respectively. Recent work shows that the actual bounds on effective coupling capacitance are –1*C_c and 3*C_c [1]. We refer to these pre-factors of (-1,3) and (0,2) as switch factors (SF). This approach is commonly limited to cases where the victim and aggressor configurations are similar – their rise times or driver strengths need to be almost identical for the switch factor to yield accurate results.

The second modeling approach to dynamic delay uses the fundamental relationship between crosstalk and dynamic delay. In [2], the authors note that neighboring wires are an added load for the victim gate and we can directly calculate the additional charge required to switch these new loads. Using the voltage glitch experienced on the victim line in the crosstalk scenario, we find an upper bound on

[1] Low-to-high transitions experience more dynamic delay since the PMOS victim device in this scenario is weaker than the NMOS aggressor.

the amount of charge needed to counteract the influence of the aggressors. In short, dynamic delay can be calculated by superimposing the crosstalk voltage glitch onto the victim switching waveform when aggressors are quiet. Note that there are limitations to this approach, as superposition may not always give the best results due to the non-linearity of the drivers. In the case of simultaneous switching, there can be an impact of victim switching on aggressor slew rate and vice versa. Expensive SPICE simulations can accurately capture such effects. Superposition approach can be used for noise aware static timing analysis by linearizing the circuit. This approach is faster than the expensive SPICE simulations and it yields good results for a range of driver and interconnect dimensions unlike SF techniques.

Recently several approaches were presented which incorporate noise into static timing analysis (STA) [3-5]. All of these methods use switching windows and switch factors. In the following section, we describe several disadvantages of switch factor based analysis. An alternative approach to noise-aware STA is presented in [7]. This paper focuses on circumventing the primary disadvantage of [7] by creating a more efficient implementation based on analytical models. Our implementation is extended to handle multiple aggressors in a noise-aware STA framework. In addition, we improve a previous noise model to enhance accuracy and present detailed results on modeling considerations that we encountered during this work.

2. NOISE AWARE TIMING ISSUES

2.1 Switch Factor Based Analysis

Most noise-aware static timing engines use switching windows to determine if noise is relevant. Typically, the timing engine will assume infinite windows [3] or use a worst-case switch factor to find an initial solution [4,5]. Iterations within the STA engine result, increasing runtime. The worse the initial estimate of dynamic delay, the worse the final solution will be in terms of either runtime or accuracy.

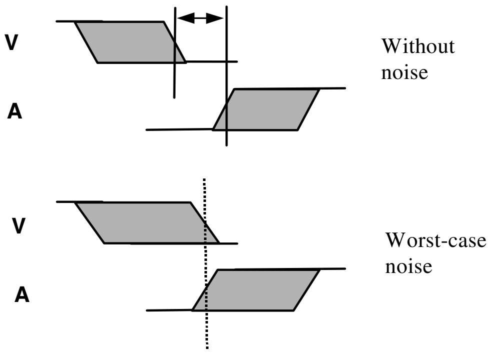

Figure 2. Assuming worst-case noise may create overlapping switching windows at fan-out nodes that do not actually exist.

Figure 2 illustrates a major problem with assuming worst-case noise in order to find noise-sensitive coupled pairs. In this figure, the victim and aggressor nets do not have overlapping switching windows when noise is not considered. Note that we are referring to windows at the fan-out nodes of the nets in question. However, if we assume that noise exists, we see that the windows will overlap – this faulty logic will cause the STA tool to incorrectly conclude that these coupled nets yield dynamic delay.

2.2 Relative Window Timing Analysis

In [7], the authors describe a novel method of dealing with dynamic delay in timing analysis. The idea is based on the observation that, while worst-case dynamic delay occurs when the aggressor and victim change nearly simultaneously, the delay is a strong function of exactly when these switching phenomena take place. When the aggressor and victim switch at time points far from one another, there is no dynamic delay impact – nominal delay is obtained. However, with a slight offset of switching events, the dynamic delay becomes less than the worst-case but still greater than zero. Traditional SF timing analysis models assume that worst-case delay applies whenever noise exists. This approximation overconstrains the design and cuts into the available timing budget; the effect becomes worse with shrinking clock periods and rising noise effects.

The concept described above is demonstrated in Figure 3, referred to as a delay change curve (DCC). In the graph, the relative signal arrival time (RSAT) is varied where RSAT is defined as the aggressor arrival time minus the victim arrival time at the gate inputs. Near RSAT=0, the maximum dynamic delay is observed. At either end of the x-axis, the dynamic delay is zero since the switching events are, in effect, independent at these points. The most interesting part of the curve is the intermediate region, where the delay is changed from its nominal value, yet it is impossible for existing modeling approaches to determine exactly how much it has changed.

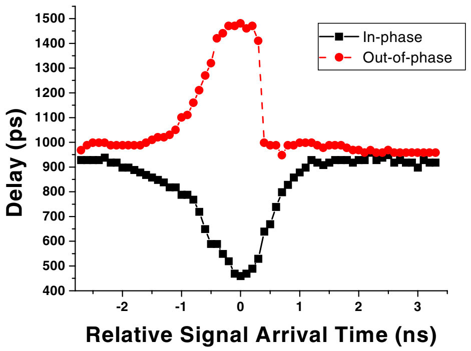

Figure 3. Measured delay change curve (DCC) for a 6 mm global line in a 0.35 μm technology.

The approach in [7] builds DCCs from circuit-level simulations during STA using a typical line length. It is unclear how results from these simulations are applied to the actual on-chip scenarios where wirelengths vary. In addition, the sheer number of simulation environments required by different interconnect configurations, drive strengths, line lengths, etc. make timing analysis based on simulation-generated DCCs impractical. However, an advantage of this approach is that by using the DCC to focus on signal arrival times, [7] reduces the conservatism shown in many noise-based timing analysis engines – worst-case noise is not always assumed. In addition, DCCs capture the possibility of dynamic delay when switching windows do not overlap while SF-based approaches cannot.

To illustrate this phenomenon, examine Figure 4. A simple inverter-based circuit has different input arrival times (AT) to the aggressor and victim gates. Here, the aggressor AT is 1.9ns (at $V_{dd}/2$) and the victim AT is 2.05ns, resulting in an RSAT of –0.15ns. Note that the waveforms propagating along the nets do not overlap, except in the last 5% of the aggressor transition. In this case, STA tools based on comparing switching windows would expect zero dynamic delay. In reality, the switching delay of the victim rises by 23%. The noise waveform on the victim arising from the earlier aggressor transition has caused the initial voltage of the victim switching event to be less than 0V. As a result, additional charge has to be supplied by the victim driver, effectively increasing the delay. In this manner, aggressor transitions well before victim switching events can contribute to delay changes. Also, the slew rate of the victim *will not be changed* under these conditions since the delay increase is only due to the initial voltage condition and not concurrent switching activity. The rise time in this case is within 1% of the scenario where the aggressor is quiet.

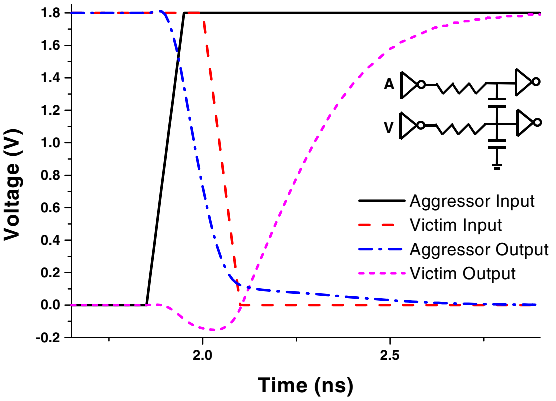

Figure 4. Noise occurs even when switching windows do not overlap due to changes in effective voltage swing.

2.3 Delay Change Curve Generation

Recent work has presented ways to measure and model the presence of dynamic delay in advanced processes. Specifically, [6] describes a set of models that relate the coupled noise waveform (the crosstalk voltage glitch) to the DCC. For example, the crosstalk noise pulse width is fundamentally related to the width of the DCC – this point is important since it gives designers and STA tools an idea of how sensitive coupled nets are to noise effects across time.

Based on simplified exponential delay and noise models, [6] demonstrated that a DCC can be quickly generated from a single circuit-level simulation used for parameter fitting to the models. However, even this single SPICE simulation is too costly for on-the-fly generation of DCCs for each net in a large design.

The remainder of this paper describes a method of improving noise-aware static timing analysis approaches. We enhance the efficiency of the relative timing window analysis methodology of [7] by analytically generating DCCs. To achieve this, we extend the work in [6] by eliminating the single SPICE run used for parameter fitting and using accurate noise waveform models to extract relevant delay model parameters. This fast generation of DCCs is extendable to multiple aggressors as well. We emphasize that any accurate and flexible analytical model can be used within this framework. To explore this concept, Section 4 describes the implications of different models and assumptions on the generated DCCs.

3. ANALYTICAL GENERATION OF DCC

In this section, we describe the overall process of quickly generating DCCs for any arbitrary driver and interconnect configuration. Since there are an unlimited number of driver sizes, gate types, interconnect topologies, fan-out conditions, etc., DCC generation must be extremely fast. To achieve this level of efficiency, we rely on analytical models to model crosstalk noise. However, analytical models are often too simple to be accurate for a wide range of such configurations. Therefore, we present several models that are useful in different scenarios and discuss applicability and limitations of each. The goal is to balance model complexity and accuracy for use in noise-aware STA. We also demonstrate in Section 3.2 that multiple aggressors are naturally integrated into the DCC generation process.

The overall flow of analytical DCC generation is shown in Figure 5. We begin with the extracted parasitics of the design, including coupling capacitances. The remainder of the flow consists of the models used in translating extracted RC parasitics to a DCC. First, the noise model is key since the noise waveform shape determines the nature of the DCC. Conceptually, the noise waveform can be

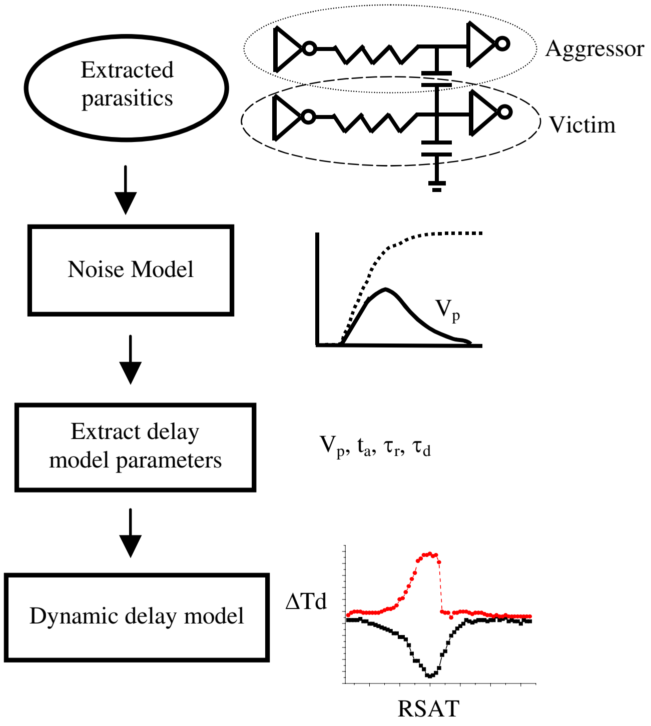

Figure 5. Overall flow to analytically generate delay change curves.

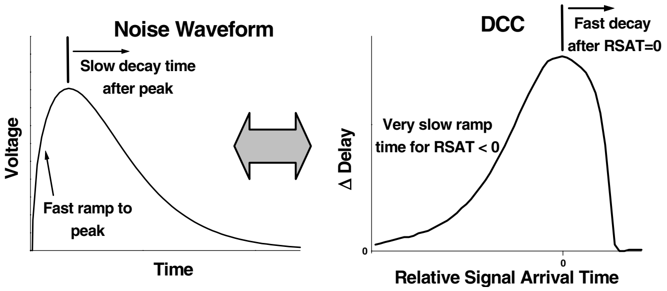

Figure 6. Qualitative description of the relationship between crosstalk noise waveform and delay change curves.

seen as a mirror image of the DCC; a slowly decaying noise spike translates to a slow ramp-up in the DCC towards the worst-case delay. Likewise, a sharp ramp-up in the noise waveform leads to a rapid decay when the RSAT becomes slightly positive. This relationship is shown graphically in Figure 6. Second, the dynamic delay model (or the translation from noise waveform to DCC) is described and various approximations are detailed. We now turn to the noise and dynamic delay models.

3.1 Analytical Modeling Flow

Our default interconnect delay model is a distributed two-pole RC line model, using an approach similar to [8]. The non-linear CMOS gate of the non-switching driver can be modeled by its effective linear resistance. However, for a switching driver, the impedance changes during switching and a single linear resistance model is not accurate. In this model, we use a ramped input voltage source to represent the driving gate. The slope of this ramp can be obtained by using any of the established C_{eff} approaches [11]. Even though these approaches are iterative, they are preferable over other approaches that try to model non-linear gates with a single resistance. In this paper, we concentrate on modeling noise and delay assuming that the slope of the ramp can be obtained accurately.

The delay waveform expression is:

$$\frac{V_{out}}{V_{dd}} = \begin{cases} \frac{t}{T_r}\left[1 - \frac{K_1 \cdot \tau_1}{t}\left(1 - e^{\frac{-t}{\tau_1}}\right) - \frac{K_2 \cdot \tau_2}{t}\left(1 - e^{\frac{-t}{\tau_2}}\right)\right], t < T_r \\ 1 - \frac{K_1 \cdot \tau_1}{T_r}\left(e^{\frac{-(t-T_r)}{\tau_1}} - e^{\frac{-t}{\tau_1}}\right) - \frac{K_2 \cdot \tau_2}{T_r}\left(e^{\frac{-(t-T_r)}{\tau_2}} - e^{\frac{-t}{\tau_2}}\right), t \geq T_r \end{cases} \tag{1}$$

The noise waveform on the victim line is described by:

$$\frac{V_{out}}{V_{dd}} = \begin{cases} \frac{t}{T_r}\left[- \frac{K_1 \cdot \tau_1}{t}\left(1 - e^{\frac{-t}{\tau_1}}\right) + \frac{K_2 \cdot \tau_2}{t}\left(1 - e^{\frac{-t}{\tau_2}}\right)\right], t < T_r \\ - \frac{K_1 \cdot \tau_1}{T_r}\left(e^{\frac{-(t-T_r)}{\tau_1}} - e^{\frac{-t}{\tau_1}}\right) + \frac{K_2 \cdot \tau_2}{T_r}\left(e^{\frac{-(t-T_r)}{\tau_2}} - e^{\frac{-t}{\tau_2}}\right), t \geq T_r \end{cases} \tag{2}$$

Details and parameters of (1) and (2) are listed in the Appendix. Equation (2) can serve as the noise model depicted in step 2 of Figure 5.

Solving for the peak noise voltage V_p in closed-form:

$$V_p = V_{dd} \cdot \frac{\tau_2 - \tau_1}{T_{r\_aggressor}} \cdot \frac{\left[K_2 \cdot (e^{T_{r\_aggressor}/\tau_2} - 1)\right]^{\tau_{2/(\tau_2 - \tau_1)}}}{\left[K_1 \cdot (e^{T_{r\_aggressor}/\tau_1} - 1)\right]^{\tau_{1/(\tau_2 - \tau_1)}}} \tag{3}$$

Here $T_{r\_aggressor}$ is the aggressor ramp rate and other parameters are given in the Appendix. In translating from crosstalk noise to dynamic delay, it is important to know the time at which the worst-case noise occurs. This is expressed as t_a:

$$t_a = \ln\left(\frac{K_1}{K_2} \cdot \frac{e^{\frac{T_{r\_aggressor}}{\tau_1}} - 1}{e^{\frac{T_{r\_aggressor}}{\tau_2}} - 1}\right) \cdot \frac{\tau_1 \cdot \tau_2}{\tau_2 - \tau_1} \tag{4}$$

With the above equations, we have the exact form of the noise waveform (of course, the exactness depends on model accuracy) and we have readily extracted two key parameters, V_p and t_a, from the waveform. The next step is to mathematically transform the noise waveform into a DCC – this is difficult to do based on a complex noise expression such as (2). Instead, we approximate the noise waveform by a simpler two-piece model with a linear ramp time (t_a) and exponential decay after the peak (τ_d). Furthermore, the without-noise victim waveform (which we need to solve for the new delay value) is approximated by a single exponential rise/fall (τ_r).

$$f(\mathrm{t},\mathrm{k}) = \begin{cases} 0, (\mathrm{t}-\mathrm{k}<0) \\ \dfrac{\mathrm{V_p}}{\mathrm{t_a}}(\mathrm{t}-\mathrm{k}), (0 \le \mathrm{t}-\mathrm{k} < \mathrm{t_a}) \\ \mathrm{V_p}\mathrm{e}^{-(\mathrm{t}-\mathrm{t_a}-\mathrm{k})/\tau_\mathrm{d}}, (\mathrm{t_a} \le \mathrm{t}-\mathrm{k}) \end{cases} \tag{5}$$

$$g(\mathrm{t}) = \begin{cases} 0, (\mathrm{t}<0) \\ \mathrm{V_{dd}}(1-\mathrm{e}^{-\mathrm{t}/\tau_\mathrm{r}}), (0 \le \mathrm{t}) \end{cases} \tag{6}$$

Here k denotes the relative signal arrival time, RSAT. These expressions can be used to generate one possible dynamic delay model, as shown in Figure 5.[2] We discuss another approximation in Section 4. We calculate V_p and t_a from Equations 3 and 4 above and then fit τ_d and τ_r in Equations 5 and 6 by comparing to the more accurate 2-pole models of Equations 1 and 2. We take this approach because the 2-pole models are considerably more complex to translate to DCCs; instead we focus on transforming the results accurately to the simpler 1-pole models of (5) and (6). After these four parameters are found, a DCC can be generated directly from the extracted RC parameters with no simulation. In the next section, a more general model is used in the same manner to handle multiple aggressors, emphasizing the model independence of the DCC generation concept.

3.2 General Models Incorporating Multiple Aggressors

In this section, we describe the extension of the above approach to practical cases with multiple aggressors. Most nets in modern designs are capacitively coupled to at least several other nets; this fact complicates timing analysis as each of the aggressors will have separate signal arrival times and will act on the victim in a distinct manner (see Figure 7). Sasaki extended his relative window analysis method to include the effects of multiple aggressors in [9]. The approach uses the absolute arrival time of the victim as a reference point so that the temporal isolation of aggressors is considered. The translation from crosstalk noise waveform to DCC described in Section 3.1 and Figure 5 is still valid in a multiple aggressor scenario so long as an approach similar to [9] is used to avoid simply summing the effect of the completely independent aggressors. The noise model, however, must be capable of handling arbitrary configurations, particularly various aggressor placements, drive directions, etc. The noise model introduced in the previous section is valid for two fully coupled lines. We now describe a general noise model and extensions we make to it in order to calculate crosstalk noise waveforms for each aggressor acting on a single victim.

The victim delay waveform when aggressors are quiet can be calculated in the traditional fashion – with coupling capacitances to aggressors viewed as capacitances to ground. Likewise, when looking at each aggressor individually, the other aggressors are considered quiet and their coupling capacitance to the victim is treated as ground capacitance (SF = 1). We use the 2-π model from [10] with modifications for the estimation of crosstalk noise. This model considers the location of coupling and can be used effectively for generic RC trees. The model also provides simple closed form expressions for noise peak and peak timing. However, [10] models the aggressor as a saturated linear ramp. In reality, it more closely approximates an exponential waveform. We extend the model to include this:

$$V_{noise}(t) = V_{dd}\left(\frac{t_x}{t_v - t_r}\right)\left[e^{\frac{-t}{t_v}} - e^{\frac{-t}{t_r}}\right] \tag{7}$$

While dealing with multiple aggressors, to solve for the victim noise waveform due to a single active aggressor we lump all the coupling capacitance due to other aggressors (quiet) at the center of the coupling. Coupling capacitances to the quiet aggressors are viewed as capacitances to ground and the resulting network is reduced to the equivalent 2-π network.

Here, t_x is the upstream resistance multiplied by the coupling capacitance, t_v is the Elmore delay of the victim net, and t_r is the time constant of the aggressor rise time (originally the rise time itself in [10]). Another problem of [10] is that it does not explicitly consider slew rate degradation along the aggressor line[3] – the aggressor ramp rate at the beginning of the line can be much different than the coupled ramp rate to the victim due to the line RC delay. We directly include this effect by dividing the aggressor line into a 2- π network and calculating the new time constant

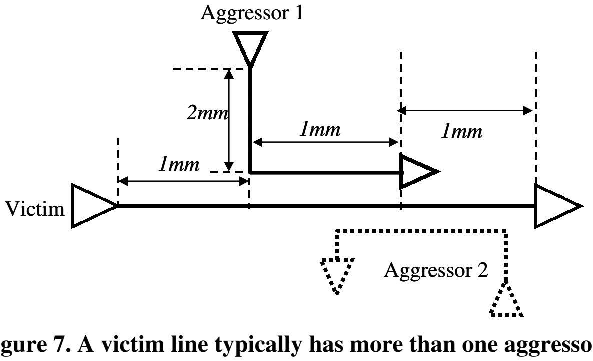

Figure 7. A victim line typically has more than one aggressor. Here, two aggressors with partial coupling complicate the timing analysis environment.

[2] The dynamic delay model resulting from (5) and (6) is used to actually plot the DCC and was introduced in [6].

[3] [10] mentions this phenomenon in passing but does not discuss its impact or describe any approaches to include the effect.

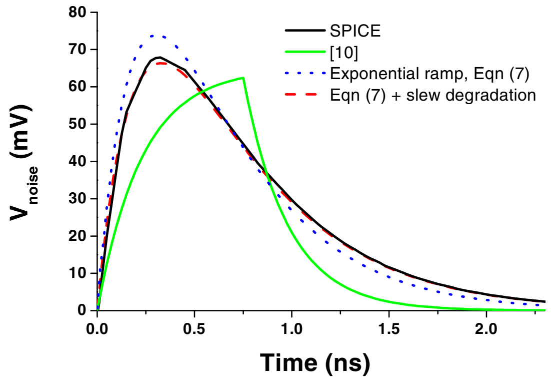

Figure 8. Model and simulation results for the victim noise waveform arising from Aggressor 1 in Figure 7.

at the coupling point. The time constant at the coupling point can be obtained by using any delay metric. The Elmore delay metric is simpler but its pessimistic results can directly translate to optimistic noise results. We use the delay metric described in [12] to calculate slew rate degradation.

In Figure 8, we show noise waveform results for [10] as well as two forms of the extended model described above for the interconnect configuration of Figure 7.[4] The inclusion of an exponential aggressor waveform is a major improvement in that the exponential rise makes noise more prominent and shifts the peak timing earlier compared to linear models. By considering slew rate degradation, the model becomes more accurate for cases where the aggressor does not couple directly at the beginning of the line, as in Figure 7.

Since the noise waveforms that result from this approach to dealing with multiple aggressors are identical in form to that from Section 3.1, there is no fundamental difference in the way the DCCs are then generated. By referencing all aggressor switching activity to the victim signal arrival time, the true impact of multiple aggressors can be determined, as shown in [9].

4. MODELING CONSIDERATIONS

In this section, we discuss the model independence of our DCC generation methodology and describe the sensitivity of our approach to model accuracy.

A major component of DCC generation is the simplification made in the noise waveform that allows a closed-form translation to DCC. We have used a model with a linear ramp to the noise peak, followed by an exponential decay to zero. A comparison of this shape with the actual noise waveform it approximates (taken from SPICE) is shown in Figure 9. As can be seen, the approximated waveform underestimates noise as it increases towards the peak because it uses a sharp peak, rather than a rounded one as seen in practice. Furthermore, the exponential tail can be fit at any particular point along the waveform and this fitting point is set at 50% of the peak value nominally. This gives a decent fit throughout the curve but results in underestimation near the peak where dynamic delay is largest. An alternative is to use an exponential rise and decay, which is also shown in Figure 9. The overall results are much better – there is some underestimation near the beginning of the noise pulse that won't strongly impact the delay calculation. Otherwise, the model fits much better than the simpler linear-exponential piecewise model of [6]. In addition, the general form of Equation (7) mirrors this waveform shape without any fitting. Our current results are based on the simpler approximation in (5) and (6) – future work will integrate the dual-exponential form into DCC generation.

A sensitivity analysis was undertaken to determine which parameters the DCC generation methodology is most sensitive to. Results for V_p are shown in Figure 10, where DCCs are generated based on the actual SPICE-extracted V_p and values with errors of ±10%. In contrast to Figure 10, deviations of ±10% in τ_d result in less than 3% error in peak noise and a half-maximum width change of ±8%.

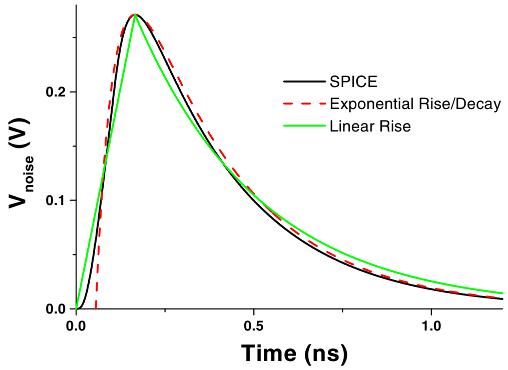

Figure 9. Exponential rise/decay assumptions fit actual noise waveforms better than simpler linear rise/exponential decay.

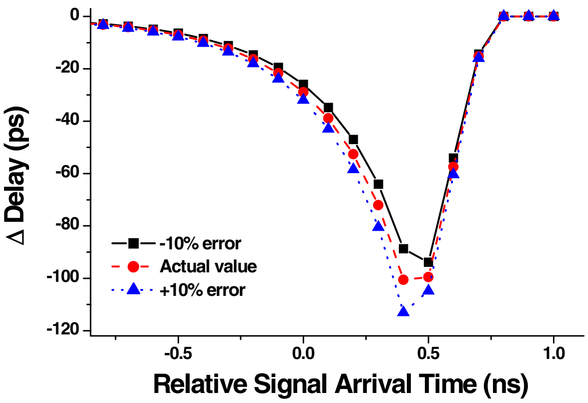

Figure 10. Error of ±10% in V_p results in major deviations in DCC shape, particularly near the peak.

[4] Linewidth and spacing are 0.35 μm, comparable to an intermediate metal level in 0.18 μm technology.

Overall, the study finds that V_p and τ_r are the most important parameters to accurately model in our approach while τ_d and t_a do not strongly impact the DCC shape. This implies that timing models are as important as noise models in determining dynamic delay. We have focused above on finding accurate noise models, but emphasize that either delay models or underlying cell timing characteristics must provide good estimates of τ_r in order to generate high-accuracy DCCs. In our simulations, we fit τ_r based on the delay model of (1).

5. DCC RESULTS

Figure 11 compares the resulting DCC (in-phase only) from several approaches for a 3 mm global net in 0.18 µm technology; full SPICE generation, single SPICE run with curve fitting [6], and the analytical approach described in Section 3.1. Results show that the analytical method is extremely accurate throughout the range of the curve. In fact, the analytical approach is comparable or superior to the results from the method using one SPICE run for equation fitting parameters. This validates the 2-pole models used to determine V_p and t_a in Section 3.1. Figure 12 shows an out-of-phase DCC generated for a 2 mm net routed on an intermediate metal layer using the improved model of Section 3.2.[5] Different line dimensions are used for victim and aggressor nets. Excellent fit is seen, with 5.5% error in half maximum width and 4.4% for peak noise. Some underestimation of delay (<10%) occurs around the peak resulting from the linear noise approximation described in Figure 9.

Tables 1 and 2 present accuracy and runtime results for the three methods of generating DCCs in Figure 11. Note that the full SPICE generation case is the method used in [7].

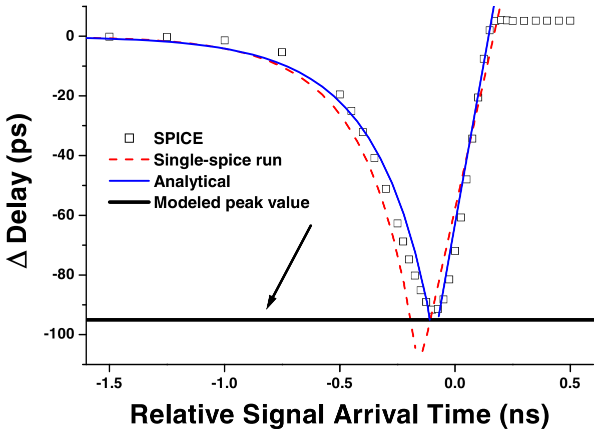

Figure 11. Resulting DCCs for full-SPICE, single-SPICE [6], analytical approaches.

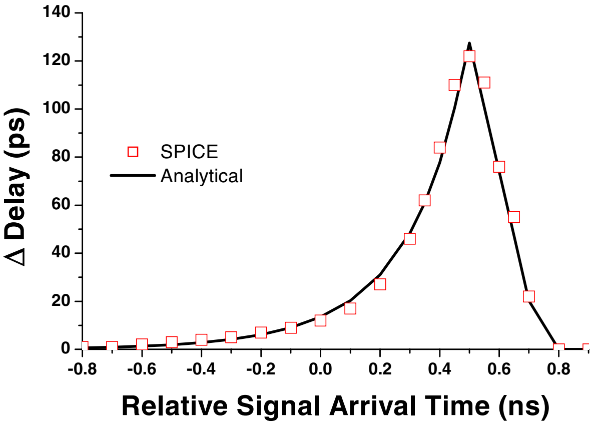

Figure 12. Out-of-phase DCC generated using SPICE and the general noise model of Section 3.2 (extended from [10]).

We examined several interconnect and driver configurations in the same 0.18 µm technology and found that the analytical approach of this work gives smaller error than [6] for nearly all cases. For a range of interconnect pitches (using 1, 2, and 3 mm M6 lines, and 0.5 and 1 mm M2 lines), we found the average error of the analytical approach to be 8% for peak delay change and 17% for DCC half-maximum width. The error is primarily due to the simple 1-pole models that we are using to drive the DCC translation. We are effectively forcing accurate data into a simplified model. These results are based on the 2-pole noise model of Section 3.1. Cases with the largest error tend to be when the noise is very small. Large noise cases such as a 3 mm M6 line or a 1 mm M2 line using minimum pitch are modeled very accurately.

Table 1. Error compared to full SPICE generated DCCs. % Error is given for peak noise / half maximum width.

Case	Line Length (mm)	[7], 1 SPICE run (%)	Analytical w/ fitting (%)
M2, P_{min}	0.5	-9.7 / -21.7	-7.1 / -26.7
M2, P_{min}	1	3.3 / -1.3	-2.9 / 0.5
M2, 2*S_{min}	1	10.8 / 0.1	3.7 / 6.5
M6, P_{min}	1	-25.1 / -42.9	-6.3 / -16.9
M6, P_{min}	3	-6.9 / -12.3	3.8 / 3.9

Table 2. Runtime in CPU seconds for three approaches to generating DCCs. Seventy distinct length and pitch combinations are simulated.

Case	Full SPICE	Single SPICE	Analytical
# of SPICE simulations / run	45	1	0
Total CPU time for 70 runs (s)	47196	1049	0.82

[5] The peak noise occurs at a large positive RSAT value here – this is due to a fast aggressor and relatively slow victim transition. Arrival times are defined by the beginning of transitions and maximum noise occurs when a victim is mid-transition and a fast aggressor then couples to it.

In addition, the runtime of the new method is much faster since simulation is completely avoided. In Table 2, we ran 70 combinations of line lengths and wire pitches and found the analytical approach took less than 1 second. Since we are still using some fitting functions (called from a perl script), the runtime is not completely negligible.

6. CONCLUSIONS

In this work, we described a fully analytical way to generate delay change curves – these DCCs can be used in timing analysis to exactly describe the exact amount of dynamic delay experienced on a net as a function of victim and aggressor arrival times. Since the number of potential aggressor/victim configurations is limitless, the efficiency of this analytical approach is necessary to make a relative window approach to STA feasible. Our results indicate that the analytically generated DCC matches the SPICE simulated curve within 5-10% on average (some cases with small levels of noise have larger errors in the half maximum width), validating the approach.

We also documented several modeling issues that arise in generating DCCs analytically. The most important conclusions from this analysis are that exponential ramps are more accurate than linear ramps in estimating noise and timing models are of equal importance to noise models in solving for DCCs. An improved general crosstalk model is developed based on [10], and shown to be useful in cases with multiple aggressors and short to intermediate line lengths. As future work, we are exploring the use of dual exponential noise waveforms for translating from an accurate noise model to a DCC analytically. Also, we are investigating the impact of victim switching on aggressor slew rate to determine how strong an effect it has on dynamic delay at the victim net. This effect can cause error in the superposition step. We are looking at possible ways to address this issue by using non-iterative switch factor based estimations to overcome uncertainty in slew rates during simultaneous switching.

APPENDIX

In Equations (1) and (2) of Section 3.1, the parameters have the following definitions:

$$K_1 = \frac{1}{2}\cdot\frac{1+R_t+C_{t1}}{\frac{\pi}{4}+R_t+C_{t1}} \quad K_2 = \frac{1}{2}\cdot\frac{1+R_t+C_{t2}/(1+\frac{2Cc}{C})}{\frac{\pi}{4}+R_t+C_{t2}/(1+\frac{2Cc}{C})} \tag{A1}$$

$$\tau_1 = (RC\cdot l^2)\cdot\left[R_t + C_{t1} + R_t\cdot C_{t1} + \left(\frac{2}{\pi}\right)^2\right] \tag{A2}$$

$$\tau_2 = (RC\cdot l^2)\cdot(1+\frac{2Cc}{C})\cdot\left[R_t + \frac{C_{t2}}{1+\frac{2Cc}{C}} + \frac{R_t C_{t2}}{1+\frac{2Cc}{C}} + \left(\frac{2}{\pi}\right)^2\right] \tag{A3}$$

and the RC parameters are defined as:

$$R_t = \frac{R_i}{R\cdot l},\quad C_{t1} = \frac{(C_{L1}\parallel C_{L2})}{C\cdot l},\quad C_{t2} = \frac{(C_{L1}+C_{L2})}{C\cdot l} \tag{A4}$$

R_i is the equivalent on-resistance of the victim driver in the linear region of operation (a technology dependent constant easily found from I-V curves), l is the line length, and R, C, and C_c are resistance, line-to-ground, and coupling capacitances per unit length (assumed to be the same for both lines in this model).

7. REFERENCES

[1] A. B. Kahng, S. Muddu, and E. Sarto, "On switch factor based analysis of coupled RC interconnects," *Proc. DAC*, pp. 79-84, 2000.

[2] F. Dartu and L. T. Pileggi, "Calculating worst-case gate delays due to dominant capacitive coupling," *Proc. DAC*, pp. 46-51, 1997.

[3] R. Arunachalam, K. Rajagopal, and L. T. Pileggi, "TACO: Timing analysis with coupling," *Proc. DAC*, pp. 266-269, 2000.

[4] P. F. Tehrani, S. W. Chyou, and U. Ekambaram, "Deep submicron static timing analysis in presence of crosstalk," *Proc. ISQED*, pp. 505-512, 2000.

[5] B. Franzini, C. Forzan, D. Pandini, P. Scandolara, and A. Dal Fabbro, "Crosstalk aware static timing analysis: a two step approach," *Proc. ISQED*, pp. 499-503, 2000.

[6] T. Sato, Y. Cao, D. Sylvester, and C. Hu, "Characterization of interconnect coupling noise using in-situ delay-change curve measurements," *International ASIC/SoC Conference*, pp.321-325, 2000.

[7] Y. Sasaki and G. De Micheli, "Crosstalk delay analysis using relative window method," *International ASIC/SoC Conference*, pp. 9-13, 1999.

[8] T. Sakurai, "Closed-form expressions for interconnection delay, coupling, and crosstalk in VLSI's", *IEEE Trans. on Electron Devices*, pp. 118-125, Jan. 1993.

[9] Y. Sasaki and K. Yano, "Multi-aggressor relative window method for timing analysis including crosstalk delay degradation," *Proc. CICC*, pp. 495-498, 2000.

[10] J. Cong, D. Z. Pan, and P. V. Srinivas, "Improved crosstalk modeling for noise constrained interconnect optimization," *ASP-DAC*, 2001

[11] J. Qian, S. Pullela, and L. T. Pillage, "Modeling the 'effective capacitance' for the RC interconnect of CMOS gates," *IEEE Trans. Computer-Aided Design*, pp. 1526-1535, Dec.1994.

[12] C. J. Alpert, A. Devgan and C. Kashyap, "A two moment RC delay metric for performance optimization," *Intl. Symp. Physical Design*, pp. 69-74, 2000