

An Efficient Algorithm for Calculating the Worst-case Delay due to Crosstalk

Venkatesan Rajappan, Synplicity Inc., Sunnyvale, CA 94086

Sachin S. Sapatnekar, ECE Department, University of Minnesota, Minneapolis, MN 55455

Abstract

Analyzing the effect of crosstalk on delay is critical for high performance circuits. The major bottleneck in performing crosstalk-induced delay analysis is the high computational cost of simulating the coupled interconnect and the nonlinear drivers. In this work, we propose an efficient iterative algorithm that avoids time-consuming nonlinear driver simulations and performs node-specific crosstalk delay analysis. The proposed algorithm has been tested over circuits in two deep submicron technologies with varying driver sizes, interconnect parasitics, signal transition times and it has been found to predict the worst-case delay to within 10 % of the actual delay.

1 Introduction

Due to scaling in process technology, coupling capacitance has become dominant and crosstalk issues have become highly critical. Crosstalk leads to two significant problems. Firstly, coupling effects may inject noise into a circuit leading to discharge of the capacitance at the output of a gate, and thereby altering functionality. This effect has been extensively researched [Che01], [Dev97], [Hey01], [Lev00], [Vit97], [She97]. Secondly, crosstalk-induced delay can critically affect circuit performance. This problem is more serious and this paper is directed towards analyzing this effect.

The coupling between interconnect lines makes it difficult to consider the effect of different aggressor drivers in an independent fashion. The traditional method of interconnect analysis that considers only one line at a time is no longer valid since the behavior of each line can depend on that of its transitive neighbors. This implies that either a large number of lines must be concurrently simulated, or that an intelligent iterative approach must be used, simulating only one line at a time. Moreover, nonlinear driver simulation greatly increases the computation time needed for analysis in either of these scenarios. Ignoring nonlinear drivers completely or modeling them with a simple linear resistance is generally known to give large errors.

Early approaches to incorporate the effects of coupling in calculating delays made use of a switch factor of [0,2] or [-1,3] for the coupling capacitance [Kah00], [Che00], [Sap00]. The worst-case bound has been found to predict overly pessimistic delay values for some cases and underestimate others [Dar97]. The exact value of the switch factor is dependent on signal polarity, driver strengths, interconnect parasitics, slew rates and arrival times and cannot be determined *a priori*. Switch factor methods do not consider many of these parameters and their accuracy for deep submicron designs is questionable.

A relative window based approach is proposed in [Sas00]. A look-up table is used to capture the change in delay with respect to the relative window for every aggressor-victim pair. In [Cho02], a model-fitting based approach is presented. Important parameters that affect the delay are identified and a model is fitted based on simulation data corresponding to each design. However, this model does not capture the dependence of crosstalk on the relative arrival times of the aggressor and the victim. This could make the model predict a crosstalk-induced delay when there is none.

An iterative C_{eff} model based approach is proposed in [Gro98]. This method captures the coupled aggressor's switching with the use of reduced order models and accounts for the additional charge due to aggressor switching. However, in addition to the coupling, the victim

driver resistance has to be accounted. Ignoring the victim driver results in large errors as shown in Fig. 1. This figure shows the results of simulating an aggressor-victim circuit and varying the aggressor alignment time with respect to the victim transition time. The noise at the victim driving point is measured and the noise peak is observed to have a variation of nearly 30%. This implies that a switching victim driver significantly alters the noise induced on the victim line.

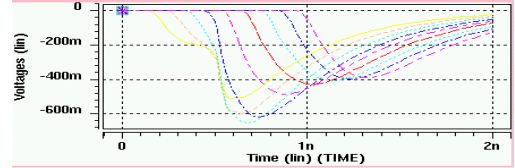


Figure 1: The variation of the noise signal with respect to aggressor alignment time.

In [Sir01], nonlinear drivers in the circuit are replaced with Thevenin equivalents and the composite waveform is obtained by linear simulation and superposition. This information is then utilized to compute a transient holding resistance either by simulation or by table look-up. However, nonlinear driver simulation is expensive and large look-up tables might be needed for varying signal slews, gate sizes and noise pulses.

In this work, we propose a novel way to analyze the worst-case delay due to crosstalk. The cyclical dependencies are handled by an iterative process that updates the driving point waveforms successively starting with noise-free waveforms. The dependence of the noise pulse on the aggressor alignment is taken into consideration by modeling the victim driver with an *alignment-time-dependent linear resistor*, thereby avoiding nonlinear driver simulations and making the method suitable for fast evaluations during design iterations.

2 Problem Definition

The problem of computing the worst-case delay due to crosstalk is defined in the following manner. The circuit under analysis is shown in Fig. 2(a), and it consists of a set of N aggressor drivers, denoted by $\{D_{A1}, D_{A2}, \dots, D_{AN}\}$, and a victim driver, denoted by D_V , driving a linear network. As in prior work [Che01], [Cho02], [Vit97], [Sir01], [Xia00], we assume that the drivers are static CMOS inverters, as shown in Fig. 2(b).

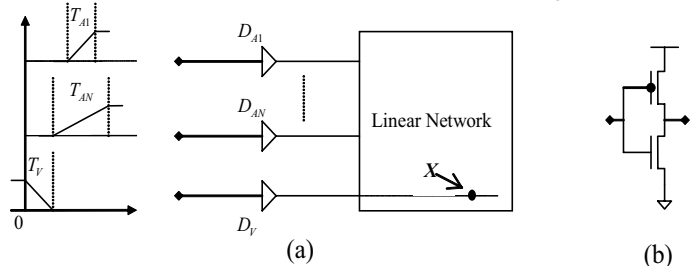


Figure 2: Typical circuit configuration for worst-case delay analysis.

The aggressor drivers are excited by rising/falling saturated ramp signals with transition times $\{T_{A1}, T_{A2}, \dots, T_{AN}\}$ and the victim driver is excited by a falling/rising saturated ramp signal with transition time T_V . The victim driver input signal starts at time 0 and stabilizes at time T_V . Assuming

that the aggressor driver input signals can arrive at any time, the *worst-case delay* for the victim output signal is to be computed at a node of interest X on the victim net. We present a solution for this aggressor alignment problem without constraints on the arrival times at the aggressor inputs, although the method can be extended to handle such constraints.

3 Issues in applying superposition

3.1 Motivation for Superposition

In this section, we discuss the results of a set of experiments designed to study the loss of accuracy caused by modeling nonlinear drivers with linear resistances.

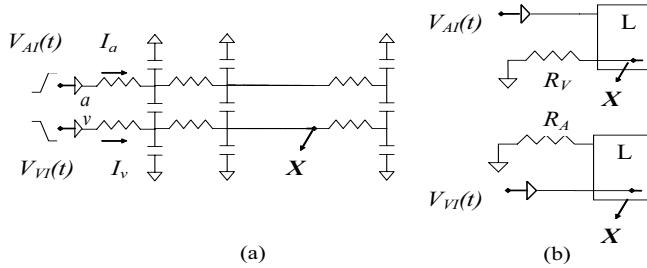


Figure 3: A two-line circuit and the superposed circuits.

Several two-line circuits, each with varying driver sizes and varying interconnect parasitics, as shown in Fig. 3(a) were considered. The aggressor is driven by $V_{Ai}(t)$, a rising/falling saturated ramp signal and the victim is driven by $V_{Vi}(t)$, a falling/rising saturated ramp signal with transition times varying from 100ps to 300ps. In each experiment, the aggressor alignment that generates the worst-case delay (WCD) at any node of interest X is determined by SPICE simulations.

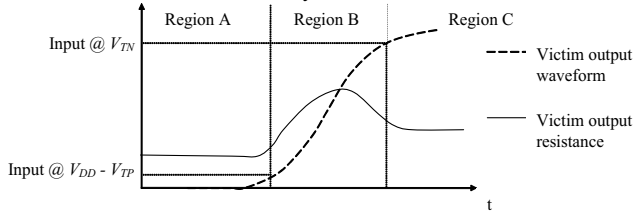


Figure 4: A graph of the victim output waveform and the output resistance.

For the worst-case delay alignment, the variation of the output resistance of the aggressor and the victim driver was measured, and a typical graph of the output resistance is shown in Fig. 4. For the purpose of explanation, let us assume that the victim driver, D_V , is excited by a falling saturated ramp signal. Regions A and C correspond to the resistance of the driver before and after the rising transition at the victim driving point, respectively, and Region B relates to the transition region resistance, and varies during the transition. The victim output resistance value corresponds to the linear region resistance of the NMOS device and the PMOS device of the driver in Fig. 2(b). The curve in Region B is determined by modeling the NMOS and the PMOS device in parallel.

The response at node X of the original circuit was compared to the responses at node X of two superposed circuits, as shown in Fig. 3(b). The first circuit is identical to the original circuit except that the victim driver is replaced by a resistance R_V . Similarly, in the second circuit the aggressor driver is replaced by a resistance R_A . The linear portion of the original network is shown as a block labeled L . Values for R_A and R_V are chosen using the following scheme. The aggressor resistance value was seen not to affect the response at node X significantly, and therefore a

single value for R_A was used. A value for R_V was chosen from Regions A, B and C. Unlike Regions A and C, the resistance value in Region B shows large variations, and therefore the experiment was performed for different resistance values from Region B and simulated with the same excitation as for the original circuit.

We found that while the delay errors of circuits with medium sized drivers using resistances from Region A or Region C were as high as 63%, the corresponding delay errors were less than 3% when resistances from Region B were used. These experiments were repeated with large and small sized drivers and similar error figures were observed. Figure 5 shows the variation in the worst-case delay for test circuits corresponding to Regions A, B and C and the original circuit in 0.18 μ technology. We observed a similar trend for test circuits in 0.13 μ technology. These results suggest that correct values can be obtained by using superposition, provided a resistance value in Region B, is chosen. This conclusion can be appreciated by noting that for a small signal such as a noise glitch, a local linear approximation, justifying the application of superposition, is possible and it translates to a constant resistance from the transition region. However, since this region experiences a wide range of resistance values, the problem is translated to that of finding an appropriate value (corresponding to the time of occurrence of the noise glitch) from this range.

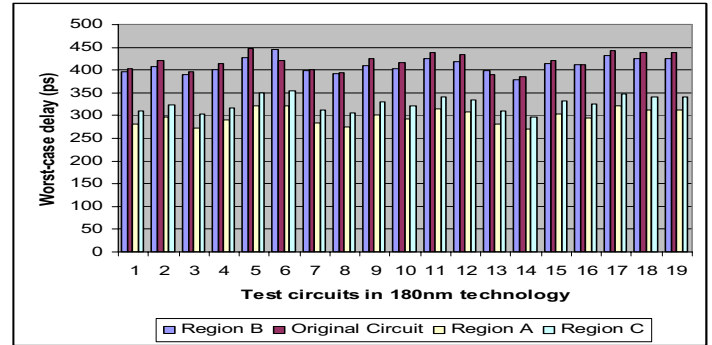


Figure 5: Worst-case delay of circuits in 180nm technology.

3.2 Finding the Worst-Case Delay Under Aggressor Alignment

The following result explains how the worst case delay over various aggressor alignments can be found by using superposition. A similar result was proved in [Gro98], but our proof appears to be simpler.

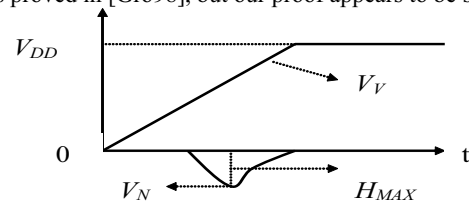


Figure 6: A noise-free victim monotone and noise signal.

Theorem 1: Consider a noise-free monotonic victim waveform V_V that starts at time 0 with slope S_V until it reaches a value of V_{DD} and an aggressor noise signal V_N of arbitrary shape with a peak height of H_{MAX} and whose initial and final values are zero, as shown in Fig. 6. If the victim waveform can be calculated as the superposed sum of these two waveforms, the worst-case 50% delay, T_{WCD} , achievable under any aggressor alignment is given by

$$T_{WCD} = \frac{(V_{DD} / 2) - H_{MAX}}{S_V}$$

Proof : Let us assume that the worst-case delay occurs at time T_{WCD} . Then at T_{WCD} , the value of the composite waveform obtained by

superposing the noise-free victim waveform and the noise signal must equal $V_{DD}/2$. Therefore,

$$S_V T_{WCD} + V_N(T_{WCD}) = V_{DD}/2$$

$$T_{WCD} = \frac{(V_{DD}/2) - V_N(T_{WCD})}{S_V}$$

Since, by definition, T_{WCD} is the largest possible delay value, we can deduce from the above relation that it corresponds to the largest value of the RHS, which in turn corresponds to the largest absolute value of V_N . By definition, the largest value of V_N is H_{MAX} and hence the proof.

Corollary 1: For a line with N aggressors, let the maximum heights of the noise signals generated by each of the N aggressors on the victim (under superposition assumptions) be denoted by, $H_{MAX1}, \dots, H_{MAXN}$, respectively. The worst-case 50% delay, T_{WCD} , achievable under any aggressor alignment is given by

$$T_{WCD} = \frac{(V_{DD}/2) - H_{MAX}}{S_V}$$

$$H_{MAX} = \sum_{i=1}^N H_{MAXi}$$

We implicitly use this result in the part of our algorithm that is described in Section 4.5 with the consideration that the noise signal varies with change in alignment.

4 Description of the Algorithm

This section presents the details of an iterative algorithm to find the worst-case delay under crosstalk. For the sake of simplicity, the algorithm is illustrated for the case of two-line circuits, as shown in Fig. 7(a), and extensions to multiple lines are discussed in Section 5. The procedure consists of six steps and an important feature of the solution is that no nonlinear driver simulations are required.

4.1 Reduction of Interconnect Parasitics to the Driving Points

As a first step towards computing the driver output resistance during transition, we reduce the network towards the driving points, as shown in Fig. 7(b), using an approach illustrated in [Sri95].

Let us consider the adjacent segments $(k+1)$ and k in the unreduced network on the left. If the admittance matrix and the impedance matrix of the k^{th} segment are denoted by Y_k and Z_k , the admittance matrix of the $(k+1)^{th}$ segment is obtained by

$$Y_{k+1} = Y_k(I + Z_k Y_k)^{-1} \quad (1)$$

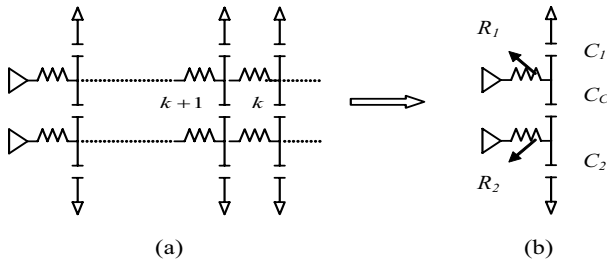


Figure 7: The multi-segment interconnect network reduced to a one-stage network.

By repeated application of (Eq. 1), the driving point admittance of the original network, $Y_{original}$, is evaluated to be of the form in (Eq. 2), where Q, R, U, X, Y are known quantities. Similarly, the driving point admittance of the one-stage reduced network, $Y_{reduced}$, is evaluated in

terms of the unknown circuit parameters R_1, R_2, C_1, C_2 , and C_C as shown in (Eq. 3). By matching respective moments, we obtain closed form equations to solve for the circuit parameters.

$$Y_{original} = \begin{bmatrix} Qs + R_s^2 & U_s \\ U_s & X_s + Y_s^2 \end{bmatrix} \quad (2)$$

$$Y_{reduced} = \begin{bmatrix} (C_1 + C_c)s + (-R_1(C_1 + C_c) - R_2 C_c^2)s^2 & -C_c s \\ -C_c s & (C_2 + C_c)s + (-R_1 C_c^2 - R_2(C_2 + C_c)s^2) \end{bmatrix} \quad (3)$$

4.2 Generation of C_{eff} Gate Delay Models

The effective capacitance (C_{eff}) technique [Aru97] models the effect of resistive shielding and derives an equivalent capacitance for the interconnect line. We use the C_{eff} technique to derive two-piece¹ noise-free aggressor and victim driving point waveforms. First, we decouple the circuits as shown in Fig. 8(a), assuming a switch factor of 1 (corresponding to a Miller capacitance of C_c , where C_c is the coupling capacitance). The switch factor is implicitly updated in Section 4.6 in the process of considering noise-included composite waveforms. Next, we derive C_{eff} models for each of the decoupled circuits, as shown in Fig. 8(b), and obtain the two-piece noise-free driving point waveforms at the aggressor and the victim outputs. A typical set of these two-piece waveforms at aggressor and victim driving points is shown in Fig. 9.

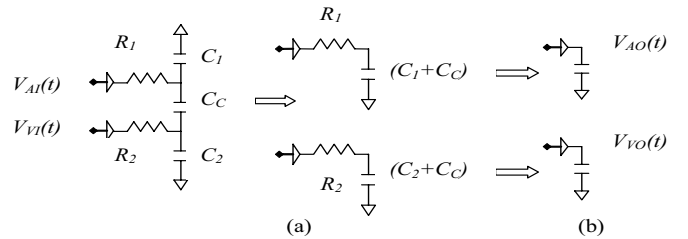


Figure 8: Decoupling of the two-line circuit into smaller circuits.

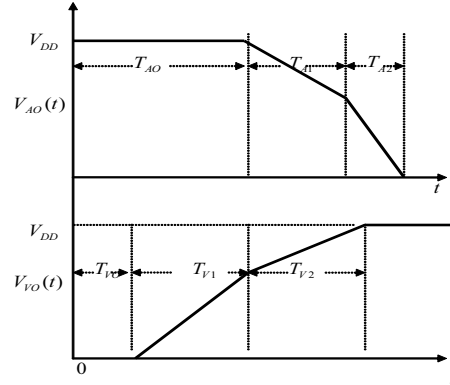


Figure 9: Typical output waveforms obtained from the C_{eff} model.

4.3 Computing the Output Resistance of the Driver

We compute the output resistance of the transiting driver from the given exact input waveforms and the approximated driving point waveforms obtained in Step 2. The region of operation of the NMOS and the PMOS device of the victim driver can be deduced from the exact input waveform and the approximated output waveform. In addition, the partial derivatives of V_{GS} and V_{DS} can be computed from the waveforms. We compute the drain current, I_{DS} , and the partial derivative of the drain current using the alpha-power law model [Sak90]. From this information, the time-varying NMOS device channel resistance, R_N , is computed by evaluating the following equation at various time points, T_X :

¹ Two-piece waveforms are essential for modeling signals in interconnect dominated stages [Aru97].

$$R_N = \left[\frac{\partial V_{DSN}}{\partial t} / \frac{\partial I_{DSN}}{\partial t} \right]_{t=T_X}$$

Similarly, the PMOS device channel resistance, R_p , is computed. Finally, the driver output resistance is computed by modeling the PMOS and the NMOS device in parallel.

$$R_D = \frac{R_p R_N}{(R_p + R_N)} \quad (4)$$

Typically, the shape of the output resistance of a transiting driver is an inverted parabola, as in Fig. 4. The values obtained from (Eq. 4) above are used to fit a quadratic function to the output resistance waveform. This quadratic function, $R_D(T_X)$, models the time-varying output resistance as a function of the relative alignment time, T_X , between the aggressor and the victim. Such a model for the output resistance is crucial since the amplitude of the noise pulse depends on the aggressor alignment as illustrated in Fig. 1.

4.4 Derivation of Aggressor and Victim Models

We now isolate the effects of the aggressor and the victim driver at the node of interest X using reduced order aggressor and victim models, as shown in Fig. 10². Realizable reduced order models are derived using AWE-based techniques [Pil90]. In order to perform node-specific analysis at any node of interest in the circuit, it suffices to compute the circuit moments only once. Unlike in Section 4.1, where we were interested in preserving only the driving point characteristics, here we would also be interested in preserving the response at the node of interest X . Therefore, it is not possible to reuse the equivalent circuit from Section 4.1, and we derive *two* reduced-order circuits, one driven by the aggressor driving point waveform, called the *aggressor model*, and one by the victim driving point waveform, called the *victim model*. In each circuit, the voltage transfer function between the driving point and X , and the driving point impedance are matched to the original circuit. The response at node X is then calculated as the sum of the responses at nodes X_A and X_V .

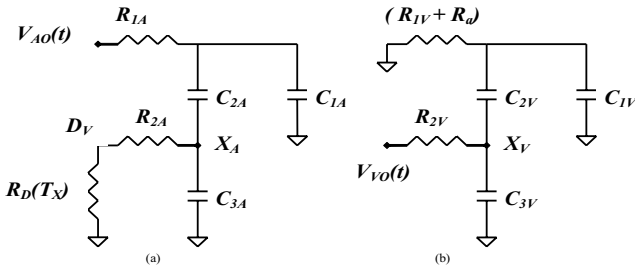


Figure 10: Circuits that model the noise contribution of the aggressor and the victim at the node of interest.

As an illustration of this process, we provide below, the details of the reduction of the original circuit in Fig. 3(a), to an aggressor model such as the one shown in Fig. 10(a). The parameters of this model are R_{1A} , R_{2A} , C_{1A} , C_{2A} and C_{3A} and the node of interest X in the original circuit corresponds to the node X_A in this circuit, and the forcing function is the approximated waveform, shown in Fig. 9, obtained in Section 4.2. Let $m_1(I_a)$, $m_1(I_v)$, $m_2(I_a)$, $m_2(I_v)$ be constants that are computed as the first and second current moments of the ports a and v in Fig. 3(a)³. Let $m_1(V_X)$ be the first voltage moment at the node of interest X . These quantities are similarly derived for the desired reduced model at the driving point and at node X_A (which corresponds to X), and a moment-matching step is

² Additional models are presented in the Appendix.

³ From experiments, we observed that using higher order moments, such as those used by models presented in the Appendix, is not always required to achieve good crosstalk delay estimates.

performed to compute the parameters of the model by solving the system of equations shown below:

$$\begin{aligned} m_1(I_a) &= C_{2A} + C_{3A}, & m_1(I_v) &= -C_{2A} \\ m_2(I_a) &= -R_{2A}(C_{2A} + C_{3A})^2 - R_{1A}C_{2A}^2 \\ m_2(I_v) &= R_{1A}C_{1A}C_{2A} + R_{1A}C_{2A}^2 + R_{2A}C_{2A}(C_{2A} + C_{3A}) \\ m_1(V_X) &= -R_{2A}(C_{2A} + C_{3A}) \end{aligned} \quad (5)$$

Having calculated the parameter values for the aggressor model, the victim driver is then replaced by the alignment-time-dependent resistance function, $R_D(T_X)$, derived in the Section 4.3.

The time-domain representation of the aggressor driving point waveform, $V_{AO}(t)$, shown in Fig. 9, is given by

$$\begin{aligned} V_{AO}(t) &= V_{D0}u(t) + K_{A1}tu(t-T_{A0}) - K_{A1}(t-T_{A0}-T_{A1})u(t-T_{A0}-T_{A1}) \\ &+ K_{A2}(t-T_{A0}-T_{A1})u(t-T_{A0}-T_{A1}) - K_{A2}(t-T_{A0}-T_{A1}-T_{A2})u(t-T_{A0}-T_{A1}-T_{A2}) \end{aligned}$$

where K_{A1} and K_{A2} are the slopes that characterize the two-piece waveform. The aggressor model in Fig. 10(a) can then easily be solved to obtain an analytical equation for the voltage at node X_A , $V_{XA}(t)$.

Similarly, an example victim model is shown in Fig. 10(b), with the circuit parameters R_{1V} , R_{2V} , C_{1V} , C_{2V} and C_{3V} and the node of interest in the original circuit is represented by X_V . The circuit parameters are obtained using a technique similar to the one illustrated for the aggressor model. The victim line is driven by the following approximated victim driving point waveform, shown in Fig. 9:

$$\begin{aligned} V_{VO}(t) &= K_{V1}(t-T_{V0})u(t-T_{V0}) - K_{V1}(t-T_{V0}-T_{V1})u(t-T_{V0}-T_{V1}) \\ &+ K_{V2}(t-T_{V0}-T_{V1})u(t-T_{V0}-T_{V1}) - K_{V2}(t-T_{V0}-T_{V1}-T_{V2})u(t-T_{V0}-T_{V1}-T_{V2}) \end{aligned}$$

As before, this system may be solved analytically to obtain the voltage at node X_V , $V_{XV}(t)$, and the details of the solution are omitted due to space constraints. Finally, the approximated voltage function at the node of interest X is given by superposing the aggressor and the victim responses at the node of interest X , as shown below:

$$V_X(t) = V_{XA}(t) + V_{XV}(t) \quad (6)$$

4.5 Aligning Aggressors for the Worst-case Delay

Our technique for calculating the aggressor alignment for the worst-case delay is based on the notion of a delay-change-curve (DCC) [Sat00], a representation of the delay as a function of the aggressor alignment. The typical shape of a DCC consists of an initial monotonically rising region that drops sharply and monotonically after reaching a peak, as shown in Fig. 11.

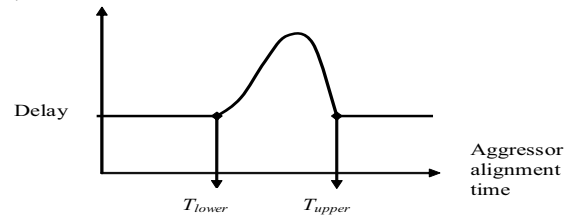


Figure 11: The typical shape of a delay-change-curve

Since our objective is to find the aggressor alignment that results in the worst-case delay, our problem reduces to that of finding the peak of the DCC. We exploit the piecewise monotone nature of the DCC and use a binary search approach, for solving this problem. For each value of the aggressor alignment T_X , the corresponding value of $R_D(T_X)$ is chosen from the curve calculated in Step 3, and a closed form solution of the reduced circuit is found from the formulae in Section 4.4. These equations are used to determine the 50% delay for the victim.

The stable values at the left end of the curve correspond to an aggressor alignment that places the noise signal on the victim before the beginning of the noise-free transition, as in Figure 12(a). The lower bound for the binary search procedure, T_{lower} , is therefore chosen as the

aggressor transition time that causes a noise signal just prior to the beginning of the victim transition. Similarly, the stable value at the right corresponds to an aggressor transition time after the end of the victim transition. For the upper bound, we use the victim waveform transition time T_V , to initially choose a conservative value of $T_{upper} = T_{lower} + 2 \times T_V$ and reduce the upper bound to the one shown in Fig. 11, by a simple binary search. Any aggressor alignment value δ between T_{lower} and T_{upper} results in a shift of δ of the noise signal on the victim waveform as shown in Figure 12(b).

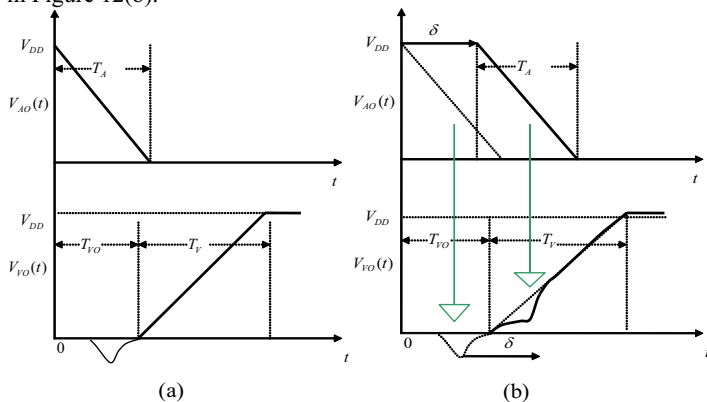


Figure 12: Effect of different aggressor alignments on the victim waveform.

As the binary search procedure converges to the peak of the DCC, implying a change in aggressor alignment, the value of $R_D(T_X)$ in the aggressor model changes resulting in varying composite victim waveform at the node of interest X. Fig. 13 shows a typical set of these waveforms obtained from our method.

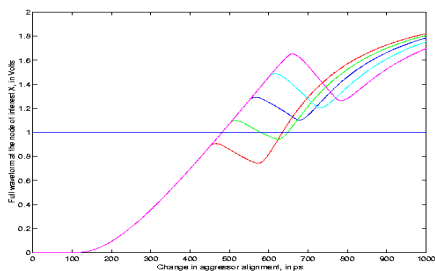


Figure 13: Voltage at the node of interest X under various aggressor alignments.

4.6 Updating the Driving Point Waveforms

In the first iteration, the worst-case delay, T_{WCD} computed in Section 4.5, is based on assuming a noise-free victim driving point waveform. However, after aggressor alignment, such an assumption is likely to be inexact, implying that the $R_D(T_X)$ waveform used in Step 4 was also inexact. Similarly, after each iteration of Steps 3 through 6, the output waveform at the victim driver is likely to change. In this step of the algorithm, we make corrections to the victim driving point waveform in the following manner. The victim driver resistance, $R_D(T_X)$, in the aggressor model in Fig. 10(a) is replaced with the resistance R_{WCD} , the resistance of the victim driver at T_{WCD} , determined from the victim output resistance graph. The noise signal at the victim driving point, D_V , is computed and a composite waveform is generated by adding the noise signal to the noise-free victim driving point transition. This implies a changed driver output resistance and necessitates recomputation of the output driver resistance in Step 3. The iteration stops when the last two values of the worst-case delay have converged to the desired accuracy.

5 Extensions to Multiple Drivers, Timing Constraints and Multiple Stages

Consideration of multiple drivers is addressed as follows:

Step 1-4, 6: Each of these steps is a straightforward extension of the procedure for the two-line case.

Step 5: The alignment of the aggressors with respect to each other and the victim waveform has to be decided. The initial alignment is performed on a one-aggressor-at-a-time basis. Each aggressor model derived in Step 4 is solved and the noise pulse at the corresponding victim driving point is computed. Each such noise waveform is aligned before the start of the victim driver output transition, as shown in Fig. 14, such that the noise peaks occur at the same time. This step is performed so that *Corollary 1* can be applied. Now, any shift in aggressor alignment suggested by the binary search procedure translates to a similar shift in all aggressor driving point waveforms.

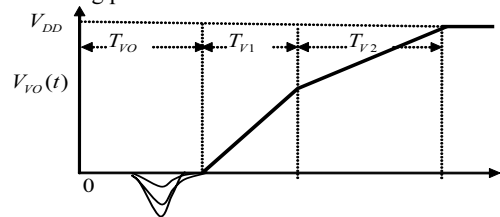


Figure 14: Aligning multiple aggressor signals before the start of the victim driver output transition.

In the presence of timing constraints, we do not have a peak-over-peak alignment. However, this extension can be used to estimate the upper bound on the worst-case delay.

The receiver can be incorporated into this framework by treating them as additional sink capacitance. It is possible to model crosstalk-affected waveforms at the input of the receiver, such as those in Figure 13, by finding equivalent waveforms using approaches similar to [Has03]. This avoids additional precharacterization and integrates seamlessly with the existing flow.

6 Experimental Results

The proposed method was implemented in 5000 lines of C code and tested on a Linux machine operating at 1800MHz. The experimental setup is as follows: Three different driver sizes belonging to 0.18 μ and 0.13 μ MOSIS technology were used in these experiments.

For the interconnect network, different parasitic sets modeling line lengths varying from 400 μ to 2000 μ , as in [Con98], were used. Two-line circuits were constructed from this set of drivers and parasitics, and the aggressor and the victim transition times were varied from 100ps to 300ps. For a chosen line length, the line resistance and the line-to-ground capacitance were uniformly distributed over ten segments. The line-to-line capacitance corresponds to spatially overlapping regions.

For each of the constructed circuits, the response at the far-end and the near-end node on the victim net were estimated by our method. The aggressor and the victim driver inputs were driven by opposite polarity signals so that the worst-case delay scenario occurs. For these circuits, we observed a maximum of 9.2 % deviation in the delay and an average error of 5.6 %. The worst-case delay of circuits in 0.18 μ and 0.13 μ technologies are shown in Figure 15 and Figure 16, respectively. The average simulation time for one HSPICE run for the worst-case delay alignment was 0.33s⁴ and the average time taken by our method to

⁴ It should be noted that to find the worst-case delay alignment, multiple enumerative HSPICE runs have to be conducted and hence the total HSPICE simulation time is typically a very large multiple of the figure mentioned here.

compute the worst-case delay was 2ms. Considering the multiple HSPICE runs needed to be performed, our method achieved an average speedup of 1000X. Typically, the method was found to converge to the worst-case delay value within 3 iterations.

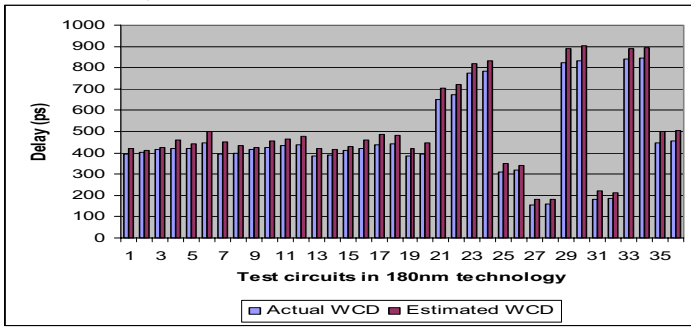


Figure 15: Actual and estimated worst-case delay values in 180nm technology.

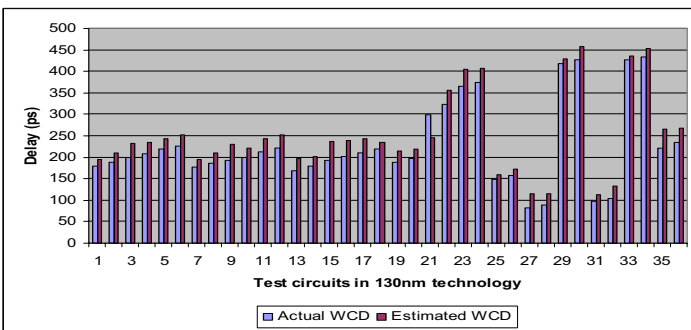


Figure 16: Actual and estimated worst-case delay values in 130nm technology.

7 Conclusions

In this paper, we addressed the problem of calculating the worst-case delay due to crosstalk at any node of interest in the circuit. The significant contribution of our work is we propose an efficient iterative algorithm that avoids nonlinear driver simulations, which are prohibitive for performing chip-level crosstalk analysis. The algorithm was tested over circuits in two deep submicron technologies with varying driver sizes, parasitics, signal transition times and was observed to predict the worst-case delay to within 10 % of the values from enumerative SPICE simulation. Future extensions of this work include incorporating timing constraints and complex gates in this framework and modifications for very large circuits.

Appendix

In our experiments, we found that for circuits with (long-aggressor, short-victim) or (short-aggressor, long-victim) the models in Fig. 17 and Fig. 18, respectively, were more accurate compared to models presented in Section 4.4. These models match upto 3 moments as opposed to the 2 moments matched by models in Section 4.4. Accordingly, we choose the particular aggressor-victim model by examining the relative lengths of the aggressor and the victim. In a fashion similar to the models solved in Section 4.4, the model parameters can be evaluated using closed-form equations.

References

- [Aru97] R. Arunachalam, F. Dartu, and L. T. Pileggi, "CMOS Gate Delay Models for General RLC Loading," *ICCD*, pp.224-229, 1997.
- [Che00] P. Chen, et al, "Miller Factor for Gate-Level Coupling Delay Calculation," *ICCAD*, pp.68-74, 2000.
- [Che01] L. H. Chen and M. Marek-Sadowska, "Aggressor Alignment for Worst-Case Crosstalk Noise," *TCAD*, pp.612-612, May 2001.
- [Cho02] S. H. Choi, F. Dartu, and K. Roy, "Timed Pattern Generation for Noise-on-Delay Calculation," *DAC*, pp.870-873, 2002.
- [Con98] J. Cong, "Challenges and Opportunities for Design Innovations in Nanometer Technologies," SRC Design Sciences Paper, 1998.
- [Dar97] F. Dartu and L. T. Pileggi, "Calculating Worst-Case Gate Delays Due to Dominant Capacitance Coupling," *DAC*, pp.46-51, 1997.
- [Dev97] A. Devgan, "Efficient coupled noise estimation for on-chip interconnects," *ICCAD*, pp.147-153, 1997.
- [Gro98] P. D. Gross, et al, "Determination of Worst-Case Aggressor Alignment for Delay Calculation," *ICCAD*, pp.212-219, 1998.
- [Has03] M. Hashimoto, et al, "Capturing Crosstalk-Induced Waveform for Accurate Static Timing Analysis," *ISPD*, pp.18-23, 2003.
- [Hey01] P. Heydari, et al, "Analysis and Reduction of Capacitive Coupling Noise in High-Speed VLSI Circuits," *ICCD*, pp.104-109, 2001.
- [Kah00] A. B. Kahng, et al, "On Switch Factor Based Analysis of Coupled RC Interconnects," *DAC*, pp.79-84, 2000.
- [Lev00] R. Levy, et al, "Clarinet : A noise analysis tool for deep submicron design", *DAC*, pp.233-238, 2000.
- [Pil90] L. T. Pillage and R. A. Rohrer, "Asymptotic Waveform Evaluation for Timing Analysis," *IEEE TCAD*, pp.352-366, Apr. 1990.
- [Sak90] T. Sakurai et al, "Alpha-Power Law MOSFET Model and its Applications to CMOS Inverter Delay and Other Formulas," *JSSC*, pp.584-594, Apr 1990.
- [Sap00] S. S. Sapatnekar, "A Timing Model Incorporating the Effect of Crosstalk on Delay and its Application to Optimal Channel Routing," *TCAD*, May 2000.
- [Sat00] T. Sato, et al, "Characterization of Interconnect Coupling Noise using In-Situ Delay Change Curve Measurements," *ASIC/SOC*, 2000.
- [Sas00] Y. Sasaki et. al, "Multi-aggressor relative window method for timing analysis including crosstalk delay degradation," *CICC*, 2000.
- [She97] K. Shepard et. al, "Global-Harmony: Coupled noise analysis for full-chip RC interconnect networks," *ICCAD*, 1997.
- [Sir01] S. Sirichotiyakul, et al, "Driver Modeling and Alignment for Worst-Case Delay Noise," *DAC*, pp.720-725, 2001.
- [Sri95] M. Sriram, et al, "Efficient Approximation of the Time Domain Response of Lossy Coupled Transmission Line Trees," *TCAD*, pp.1013-1024, Aug. 1995.
- [Vit97] A. Vittal, et al, "Crosstalk in VLSI Interconnections," *TCAD*, pp.1817-1824, Dec. 1997.
- [Xia00] T. Xiao and M. Marek-Sadowska, "Worst Delay Estimation in Crosstalk Aware Static Timing Analysis," *ICCD*, 2000.

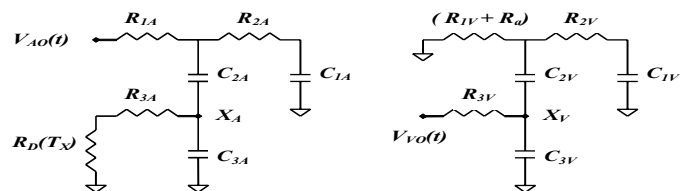


Figure 17: Aggressor and victim models for a circuit with longer aggressor line and shorter victim line.

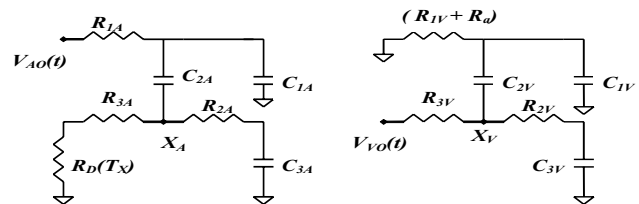


Figure 18: Aggressor and victim models for a circuit with shorter aggressor line and longer victim line.