Crosstalk Analysis using Reconvergence Correlation

Sachin Shrivastava, Rajendra Pratap, Harindranath Parameswaran, and Manuj Verma

Cadence Design Systems, India Pvt. Ltd.
Noida Special Economy Zone, Noida, India-201305
Tel : +91-120-2562842 , Fax : +91-120-2562231
e-mail: sachins@cadence.com,pratap@cadence.com,harindra@cadence.com
and mverma@cadence.com

Abstract— In the UDSM era, crosstalk is an area of considerable concern for designers, as it can have a considerable impact on the yield, both in terms of functionality and operating frequency. Methods of crosstalk analysis are pessimistic in nature and the effort is ongoing to come up with techniques that make the analysis as realistic as possible. Using information from timing analysis is one such technique where we use data about overlap in switching among nets to identify those that can potentially switch together. Existing techniques tend to look at the set of a victim and associated aggressor nets in isolation, and select a subset of aggressors based on the absolute timing windows of these nets, thus ignoring the information associated with the fanin of these nets. In reality, however, some of these nets may never switch together because the reconvergence of those nets has not being factored in. Ignoring this correlation can cause false failures being flagged, leading to increased design cycles and conservatism in the design. We propose a technique where the correlation due to reconvergence can be captured in terms of relative switching windows [1]. We apply this technique to real designs and show that this leads to more realistic analysis for crosstalk, and that we can see a reduction in the number of violations reported. We also analyze the effective of the method statistically.

I. Introduction

With shrinking process dimensions, and incresingly dense designs, crosstalk has been an area of increasing concern to ASIC designers over the past few years. Crosstalk can cause quiescent nets to glitch leading to functional failures, and can cause switching nets to speed up and slow down (depending on the relative directions of switching) thus potentially leading to timing violations in the design. Crosstalk is extremely complex to analyze accurately, so existing techniques make several approximations that can speed up the analysis. These include usage of methods like constraint propagation, generation of noise models for cells, and usage of timing information. All of these techniques tend to make the result of analysis pessimistic, and can lead to the flagging of false violations, where the analysis tool shows a particular net or path to be failing, when in reality the design may perform correctly under the given operating conditions. This can lead to longer design cycles when the designer attempts to fix or verify the failure, and can also lead to overly conservative designs. One common technique used in analysis is to use the timing windows of the set of aggressor net(s) and of the designated victim net so as to find the worst potential noise [1][2][3][4][5]. In crosstalk glitch analysis (where the victim net is static), we look at the set of aggressors to see the strongest (in terms of glitch generated at the victim net receiver input) subset of aggressors that can switch in a common window. Similarly for finding the effect of crosstalk on delay, we look at a scenario where the victim net is switching together or against a set of aggressors, and try to find the combination where the victim signal switches the slowest (victim switching against the aggressor nets) or the fastest (victim switching with the aggressor nets). In general, this problem can be looked at in terms of taking a set of nets and trying to identify a set that maximizes some effect of interest - in this case glitch on victim net or timing change in the victim signal. At a high level, this problem can be decomposed into two parts, one where we identify all possible sets of aggressors that can switch together, and then trying to see which of these sets can have the maximum impact.

Existing approaches take the absolute timing windows of each of these aggressors, and either consider overlap, or take the overlap at the tap point. However, these approaches take the timing windows to be independent of each other. This however, can lead to pessimistic results because it does not consider the case where these nets may not be able to switch together due to reconvergence correlation between the nets. By reconvergence we mean the scenario where nets diverge from a common point and later in the fan-out cone there is need to find if these nets can switch together

Consider the case shown in Fig. 1. Here we have assumed

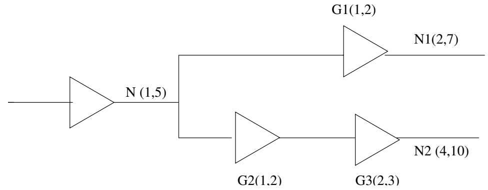

Fig. 1. Absolute timing windows

[1] The term "Relative Window" is also used in [8], to avoid any confusion we would like to mention that relative timing windows discussed in this paper is totally different from [8], and should not be confused with the concept defined there.

0-7803-9451-8/06/$20.00

zero net delays without loss of generality since nets can always be replaced by buffers of equivalent delay. In this figure timing windows of net N is (1,5) and timing windows of net $N1$ and $N2$, can be found by adding gate delays $G1$ and $G2 + G3$ respectively. It therefore appears that timing windows of net $N1$ and $N2$ are overlapping, hence in conventional crosstalk analysis they are considered to be switching together. If these two nets are considered as attackers to a common victim net, analysis for worst case crosstalk effect on the victim net will take in the scenario where both of these nets attack the victim together. However, note that if net N switches at time t, net $N1$ switches in time $[t + 1, t + 2]$, while net $N2$ switches during $[t + 3, t + 5]$. Hence, in reality $N1$ and $N2$ can never switch together (except for the case when difference between two delays is more than time period, and transition of one net in n^{th} cycle can overlap with transition of other net in $n + 1^{th}$ cycle.), and therefore cannot attack the victim net simultaneously. This leads to pessimistic analysis, and can result in a false failure being flagged at the victim net.

This pessimistic analysis happens because the technique does not consider the fact that in this case net $N1$ and $N2$ are correlated due to reconvergence. By reconvergence we mean that nets which diverge from a common point and which later are grouped together in the same set for analysis. In this case it applies to a set of nets which are being examined for a common switching windows. Since these nets diverged from some common point they will exhibit some correlation in their timing windows. Any analysis that ignores this correlation will tend to be inaccurate.

In this paper we define timing windows with reference to the diverging node in the fan-in cone of the net. We propose a method to perform crosstalk analysis using the relative timing windows which utilizes the re-convergence correlation, and thus addresses this source of pessimism present in the current approach.

We also examine this problem analytically, and show using probabilistic models that the probability of a violation being flagged reduces using this method.We have also shown some data on real designs which shows the effectiveness of the method proposed in the paper.

II. Relative Timing Windows

Let (t^N_{min}, t^N_{max}) be the timing window of a net N, such that net N can switch during time t^N_{min} to t^N_{max}. Similarly (d^P_{min}, d^P_{max}) represents delay of a path P, where d^P_{min} is the minimum delay and d^P_{max} is maximum delay.

In Fig. 2, the timing windows of net N can be found by adding delays to the timing windows.

$$TW(N) = \bigcup_{i=1}^{n} TW(Ni) \oplus D(i) \tag{1}$$

where $TW(Ni) = (t^{Ni}_{min}, t^{Ni}_{max})$ is the timing window of net Ni, and $D(Ni \to N) = (d^{Ni\to N}_{min}, d^{Ni\to N}_{max})$ is the delay from net Ni to N. Here the *and* operation $\oplus$ is defined as $(a, b) \oplus (c, d) = (a + c, b + d)$.

In the absolute timing windows approach, we lose the relative information of the delays between net N and Ni due to this addition of delays to timing windows.

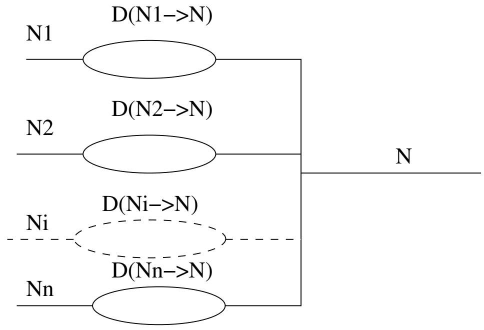

Fig. 2. Relative Timing windows

In the relative timing windows based approach proposed here, instead of adding delays to timing windows, the delays and timing windows of fan-in nets are preserved. Timing window of net N with reference to net Ni can be represented as $TW(N/Ni) = \{TW(N), D(Ni \to N)\}$. Hence,

$$TW(N) = \bigcup_{i=1}^{n} TW(N/Ni) \tag{2}$$

If net N has relative timing windows with reference to $N1$, and net $N1$ has timing windows with reference to net $N2$, timing windows of net N can be found with reference to $N2$. If,

$$TW(N/N1) = \{TW(N1), D(N1 \to N)\} \tag{3}$$

$$TW(N1/N2) = \{TW(N2), D(N2 \to N1)\} \tag{4}$$

Thus, the timing windows of net N with reference to $N2$ will be,

$$TW(N/N2) = \{TW(N2), D(N2 \to N1) \oplus D(N1 \to N)\} \tag{5}$$

III. Relative Timing Windows Based Crosstalk Analysis

In crosstalk analysis (noise or delay), timing windows are used to find if two nets can switch together or not. As we saw in ection I that nets $N1$ and $N2$ in Fig. 1 are considered to be switching together using common timing windows based approach.

If we use relative timing windows in this case, net $N1$ and $N2$ will have relative timing windows $\{TW(N), (1, 2)\}$ and $\{TW(N), (3, 5)\}$ respectively. Based on relative timing windows, timing windows of net $N1$ and $N2$ with reference to net N are $(1, 2)$ and $(3, 5)$, hence from use of relative timing windows, it is clear that net $N1$ and $N2$ can never switch together. The approach suggested here applies only to the nets constrained by synchronous clocks, since nets constrained by asynchronous clocks will be uncorrelated and can switch at any time.

Consider the case where we need to find out if two aggressor nets $A1$ and $A2$ can switch together as part of our analysis. Let

$\Phi(N)$ be the set of nets at all latest divergence points in the fan-in cone of net N. Then the relative timing windows of net $A1$ and $A2$ can be represented as,

$$TW(A1) = \bigcup_{i \in \Phi(A1)} \{TW(Ni), D(Ni \rightarrow A1)\} \quad (6)$$

$$TW(A2) = \bigcup_{i \in \Phi(A2)} \{TW(Ni), D(Ni \rightarrow A2)\} \quad (7)$$

Note that since aggressors $A1$ and $A2$ are constrained by synchronous clocks, they will always have common divergence points in the fan-in cone. If we trace back in the fan-in cone, we can find latest common points in the fan-in cones using standard graph algorithms. Let the set of latest common point in these fan-in cones be $\Phi(A1, A2)$. Using equations (3,4) we can represent timing windows of net $A1$ and $A2$ with reference to nets in set $\Phi(A1, A2)$.

$$TW(A1) = \bigcup_{i \in \Phi(A1,A2)} \{TW(Ni), D(Ni \rightarrow A1)\} \quad (8)$$

$$TW(A2) = \bigcup_{i \in \Phi(A1,A2)} \{TW(Ni), D(Ni \rightarrow A2)\} \quad (9)$$

While finding the overlap relationship of net $A1$ and $A2$, the first term in the relative timing windows (i.e. $TW(Ni)$) can be ignored since it is common for both nets. Now, based on delays $D(Ni \rightarrow A1)$ and $D(Ni \rightarrow A2)$, it can be found whether nets can switch together or not. It is immediately apparent that since delays $D(Ni \rightarrow A1)$ and $D(Ni \rightarrow A2)$ will be significantly less wider than the timing windows, this approach reduces the pessimism caused in the standard approach.

So far we did not consider the direction of switching (rise or fall). Due to different rise and fall delays, rise and fall timing windows can be different, and hence we need to find the rise and fall relative timing window of a net. In the conventional analysis we have absolute timing windows available for all the nets, hence we need to find a way to calculate relative timing windows using the absolute timing windows.

Let the timing window at a divergence point be (R_1, R_2) for rise transition and (F_1, F_2) for fall transition. Also let the delays of the path from this divergence point to the aggressor net under consideration be (r_1, r_2) and (f_1, f_2) for rise and fall transition respectively. Now the timing window at aggressor will depend on the unateness of the path. Hence, the rise and fall timing windows at the aggressor net will be,

- For a non-inverting path

$$(R_1 + r_1, R_2 + r_2), (F_1 + f_1, F_2 + f_2) \quad (10)$$

- For an inverting path

$$(F_1 + r_1, F_2 + r_2), (R_1 + f_1, R_2 + f_2) \quad (11)$$

- For a path which can be both inverting and non-inverting.

$$(min(R_1, F_1) + r_1, max(R_2, F_2) + r_2),$$
$$(min(R_1, F_1) + f_1, max(R_2, F_2) + f_2) \quad (12)$$

If the timing windows at the divergence point are (R_1', R_2') for rise transition and (F_1', F_2') for fall transition, the relative timing windows will be,

- For a non-inverting path

$$(R_1' - R_1, R_2' - R_2)(F_1' - F_1, F_2' - F_2) \quad (13)$$

- For an inverting path

$$(R_1' - F_1, R_2' - F_2), (F_1' - R_1, F_2' + R_2) \quad (14)$$

- For a path which can be both inverting and non-inverting

$$(R_1' - min(R_1, F_1), R_2' - max(R_2, F_2)),$$
$$(F_1' - (min(R_1, F_1), F_2' - max(R_2, F_2) \quad (15)$$

We now propose a crosstalk noise analysis method using timing windows below

1. Perform Crosstalk analysis using conventional approach, and find list of violating nets.
2. For each violating net in the design.
   (a) Find all aggressors of this net.
   (b) For each synchronous group of aggressors
      i. Find common divergence points of all the nets in the group.
      ii. Find relative timing windows using equations (13-15)
      iii. Use relative timing windows to find which nets in the group can actually switch together, and modify the aggressor groups.
   (c) Based on the new groups, find glitch caused by the worst group.

Note that in the case of crosstalk delay analysis, we need to compute sets of aggressors that switch with victim net. This means that while finding common divergence point, victim net is also to be considered along with the aggressors.

IV. Analytical study using probabilistic models

In this section we find the effectiveness of the proposed approach by finding the probability that this approach will reduce the glitch on a given net. Consider a victim net which is affected by N aggressors. For simplicity and without loss of generality we assume that all the aggressors are constrained by clocks of same time period T (This assumption can be relaxed by considering LCM of all time periods and then we can represent discontinuous timing windows). In the analysis we are assuming that all switching instances are uniformly distributed in the time period T (Though due to this assumption, the analysis may not be very realistic, but it will be indicative since we are using this analysis only to find the change in violations using our approach). If the width of the timing window of ith aggressor is τ_i, the probability of all the nets switching together is (Refer Appendix A, [7])

$$P_{old} = \left(\prod_{i=1}^{N} \frac{\tau_i}{T} \right) \left(\sum_{j=1}^{N} \frac{T}{\tau_j} \right) \tag{16}$$

If we use our relative timing windows based approach , and if the width of timing window at the common divergence point is d, the probability will be

$$P_{new} = \left(\prod_{i=1}^{N} \frac{(\tau_i - d)}{T} \right) \left(\sum_{j=1}^{N} \frac{T}{(\tau_j - d)} \right) \tag{17}$$

Hence, given that N nets are switching together using conventional analysis, probability that they will not be considered switching together using our approach,

$$P = \frac{P_{new}}{P_{old}} \tag{18}$$

If we assume that widths of all the timing windows are same τ, the probability would be

$$P = \left(1 - \frac{d}{\tau} \right)^{N-1} \tag{19}$$

This shows that by using proposed approach glitch on the net will decrease with a probability P.

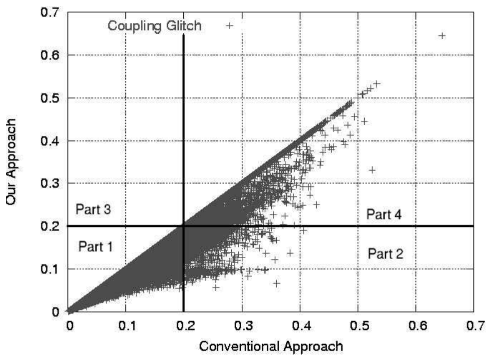

Fig. 3. Coupling glitch comparison

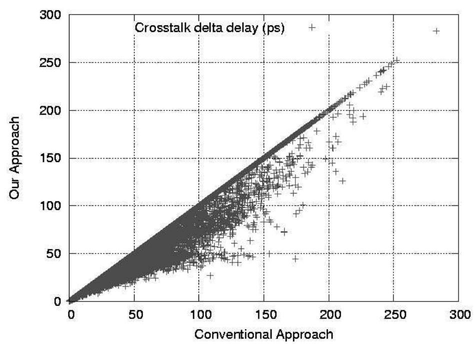

Fig. 4. Crosstalk delay comparison

V. Experimental Results and Conclusions

We used this approach on a 65nm design using the method mentioned in section III. Fig. 3 shows the scattered plot comparison of coupling glitch obtained using our method with the existing approach. For simplicity we assume that all the cells have a threshold of 0.2V and glitch is not propagated through cells [3][6]. Though results with these assumptions may not be very accurate but will be indicative since coupling glitch has direct impact on number of violations.

We divide the plot into four partitions as following,

Part 1 : Number of nets below threshold (using existing and proposed approach) = 139634

Part 2 : Number of nets above threshold (using existing approach) and below thresholds (using proposed approach) = 2053

Part 3 : Number of nets below threshold (using existing approach) and above thresholds (using proposed approach) = 0

Part 4 : Number of nets above threshold (using existing and proposed approach) = 14906

This means that Part 2 will have nets which were reported as violation using existing approach, but are filtered using proposed approach. Part 3 will have nets which were not reported as violation using existing approach, but are reported as violation using the proposed approach, but since this approach does not add any pessimism, number of nets in this part will always be 0.

Hence earlier, there were 16959 (Part 2 + Part 4) violations, and our approach filtered 2053 (Part 2) false violations, and now there are only 14906 (Part 3 + Part 4) potential violations left.

It is clear from the results that this approach reduces pessimism in the coupling glitch computation, and reduces the number of violations, making the analysis more accurate.

Fig. 4 shows similar comparison for crosstalk delay analysis. Here we have compared the delta delay at the victim net due to crosstalk. In crosstalk delay analysis victim timing window is also considered [4][5], hence this approach also removes the pessimism in victim-aggressor overlap, as compared to noise analysis where only aggressor-aggressor overlap is considered for pessimism removal.

VI. Conclusions

In this paper we have proposed the concept of relative timing windows and a method to perform crosstalk analysis using relative timing windows. As compared to conventional analysis where nets are considered independent of each other, our method utilizes the reconvergence correlation between two nets, and reduces the pessimism present in the conventional approach.

We analyze the effectiveness of this approach statistically using simplified probabilistic models. Experimental results have been presented that helps in showing the effectiveness of this approach considering both crosstalk glitch and delay analysis.

Appendix

A. Probability of aggressors switching together

In this section we find the probability of N nets switching together, if there timing windows of width $(\tau_1, \tau_2, \ldots, \tau_N)$ are uniformly distributed in the period T. Since switching will repeat after time period T, we consider time period T as the periphery of circle, and timing windows uniformly distributed around the circle.

We consider the discrete problem first by dividing the time period T in X equal increments, then we can find solution for continuous case by making $X \to \infty$.

Now timing windows will have widths $m_1, m_2, \ldots, m_N$ increments, where $m_i = (\tau/T)X$. All these timing windows can be placed in X^N number of ways. In order to find the probability of all of them switching together, we need to find the number of combinations which contains at least one increments of overlap between all the timing windows.

Each combination of overlap can be translated into X different positions, hence we need to first find number of overlapping combination for just one interval. If we consider one fix interval in the duration, number of combinations in which all timing windows will contain this interval will be $m_1 m_2 \ldots m_N$. But due to the translation, overlap of two increments is counted twice, similarly overlap of three increments is counted thrice and so on. Hence if we fix two particular consecutive increments in the duration, number of combinations which overlap will be $(m_1 - 1)(m_2 - 1)\ldots(m_N - 1)$. This counts overlap of two increments once, overlap of three increments twice and so on. Hence if we subtract this product from the previous product, we will get the number of combinations which counts each overlap exactly once.

$$m_1 m_2 \ldots m_N - (m_1 - 1)(m_2 - 1)\ldots(m_N - 1)$$

Hence probability of N nets switching together,

$$P = \lim_{X \to \infty} \frac{X[m_1 \ldots m_N - (m_1 - 1)\ldots(m_N - 1)]}{X^N} \quad (20)$$

Since $m_i = (\tau_i/T)X$, all the term in numerator of degree less than N drop out, and the probability of N nets switching together for continuous case is,

$$P = \left(\prod_{i=1}^{N} \frac{\tau_i}{T}\right)\left(\sum_{j=1}^{N} \frac{T}{\tau_j}\right) \quad (21)$$

References

[1] L.H. Chen and M. Marek-Sadowska, *"Aggressor Alignment for Worst-Case Crosstalk Noise"*, IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, vol20, no. 5, May 2001, pp. 612-621.

[2] P. Chen and K. Keutzer, *"Toward True Crosstalk Noise Analysis"*, Proc. Intl. Conf. Computer-Aided Design, 1999.

[3] K.L. Shepherd, V. Narayanan, P.C. Elmendorf, and G.Zheng, *"Global Harmony: Coupled Noise Analysis for Full-Chip RC Interconnect Analysis"*, Proc. Intl. Conf. Computer-Aided Design, Nov. 1997, pp. 139-146.

[4] G. Yee, R. Chandra, V. Ganesan, and C. Sechen, *"Wire Delay in the Presence of Crosstalk"*, Proc. IEEE/ACM Intl. Workshop on Timing Issues in the Specification and Synthesis of Digital Systems, pp. 170-175,1997.

[5] P.D. Gross, R. Arunachalam, K. Rajagopal, and L.T. Pileggi, *"Determination of worst-case aggressor alignment for delay calculation"*, Proc. Intl. Conference Computer-Aided Design, Nov. 1998, pp. 212-219.

[6] V. Zolotov, D. Blaauw, S. Sirichotiyakul, M. Becer, C. Oh, R. Panda, A. Grinshpon, and R. Levy, *"Noise propagation and failure criteria for VLSI designs"*, Proc. IEEE/ACM Intl. Conf. Computer-Aided Design, 2002.

[7] A. Papoulis, *"Probability, Random Variables, and Stochastic Processes"*, 2nd ed. New York: McGraw-Hill, 1984.

[8] Y. Sasaki and G. De Micheli, *"Crosstalk Delay Analysis using Relative Window Method"*, ASIC/SoC Conference, 1999.