CROSSTALK NOISE ANALYSIS FOR NANO-METER VLSI CIRCUITS

by

Ravikishore Gandikota

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering)
in The University of Michigan
2009

Doctoral Committee:

Professor David Blaauw, Chair
Associate Professor Igor L. Markov
Associate Professor Dennis Michael Sylvester
Assistant Professor Amy Ellen Mainville Cohn

To my family and friends for always being a constant source of inspiration

ACKNOWLEDGEMENTS

I would like to express my gratitude for every individual who has helped me during my stay here as a graduate student at the University of Michigan.

I would like to thank my research advisor Prof. David Blaauw for his invaluable guidance and support throughout my doctoral studies. I would also like to thank Prof. Dennis Sylvester who was like a second research advisor to me throughout my graduate studies. I would also like to thank the other members of my dissertation committee, Prof Markov and Prof Cohn for their invaluable feedback and suggestions to greatly improve the quality and content of my dissertation thesis.

I would like to thank my colleague Kaviraj Chopra who helped me in numerous ways in my research work. I would also like to thank my senior graduate students (Sanjay, Vishvesh, Brian, Carlos, Sarvesh and Ashish) at for supporting and assisting me greatly during my formative years here at the University of Michigan. I also thank all the students in the CAD group for the discussions and feedback on my research.

I will take this opportunity to thank Shantanu for being a very understanding roommate. I also thank Sudherssen, Vineeth and Mingoo for being such nice project partners in the graduate VLSI courses. I thank my brother Arun for always being a thoughtful and a caring person. Last but not the least, I thank my parents for always encouraging and guiding me in life.

TABLE OF CONTENTS

DEDICATION . . . . . **ii**

ACKNOWLEDGEMENTS . . . . . **iii**

LIST OF FIGURES . . . . . **vii**

LIST OF TABLES . . . . . **x**

LIST OF APPENDICES . . . . . **xi**

ABSTRACT . . . . . **xii**

CHAPTER

I. Introduction 1

1.1 Static Timing Analysis 1
1.2 Crosstalk Noise 4
1.3 Crosstalk-Noise Filtering 6
1.3.1 Spatial Filtering 7
1.3.2 Functional Filtering 7
1.3.3 Temporal Filtering 8
1.4 Miller-Coupling Capacitance 9
1.5 Crosstalk-Noise Analysis 11
1.5.1 Noise Analysis with Linear Drivers 11
1.5.2 Noise Analysis with Nonlinear Drivers 14
1.5.3 Impact of Coupling Noise on Design Performance 16
1.6 Thesis Overview and Key Contributions 20

II. Victim Alignment in Crosstalk-Aware Timing Analysis 23

2.1 Problem Description 26
2.2 Victim Alignment for Linear-Driver Models 30
2.3 Victim Alignment for Nonlinear-Driver Models 33
2.3.1 Properties of Nonlinear Drivers 33
2.3.2 Worst-Case Victim Alignment 35
2.3.3 Worst-Case Victim-Receiver Output Alignment 44
2.3.4 Monotonic-Input Transitions 45
2.4 Victim Alignment for Multiple Aggressors 46
2.5 Experimental Results 48
2.5.1 HSPICE Simulations 48
2.5.2 Delay-Noise Analysis 49
2.6 Summary 52

III. Worst-Case Aggressor-Victim Alignment with Current-Source Drivers . . 53

3.1 Problem Description . . . 56
3.2 Cumulative Gate Overdrive Voltage . . . 60
3.3 Worst-Case Aggressor Alignment . . . 63
3.4 Experimental Results . . . 68
3.5 Summary . . . 72

IV. Top-k Aggressors in Noise Analysis . . . 73

4.1 Problem Description . . . 77
4.1.1 Challenges in Computing the Top-k Aggressors' Set . . . 77
4.2 Proposed Approach . . . 79
4.2.1 Pseudo-Input Aggressors . . . 81
4.2.2 Aggressor Dominance . . . 82
4.2.3 Algorithm to Compute the *Top-k* Aggressors' Addition Set . . . 85
4.2.4 Algorithm to Compute the *Top-k* Aggressors' Elimination Set . . . 88
4.3 Experimental Results . . . 89
4.4 Summary . . . 91

V. Modeling Crosstalk in Statistical Static Timing Analysis . . . 93

5.1 Problem Description . . . 97
5.2 Statistical-Delay Noise . . . 100
5.2.1 Correlations in Delay Noise . . . 100
5.2.2 Canonical Delay-Noise Distribution . . . 103
5.3 Delay Noise in SSTA . . . 105
5.3.1 Delay-Noise Distribution using Skew Window . . . 105
5.3.2 Multiple Skew Windows . . . 107
5.4 Experimental Results . . . 108
5.4.1 Results on Benchmark Circuits . . . 109
5.5 Summary . . . 112

VI. Pessimism Reduction with Path-Based Statistical Noise Analysis . . . 113

6.1 Proposed Approach . . . 118
6.1.1 Path Delay-Noise Distribution . . . 118
6.1.2 Statistical Noise Analysis of the Victim Stage . . . 121
6.1.3 Worst-Case Alignment of Top-k Aggressors . . . 125
6.2 Experimental Results . . . 126
6.3 Summary . . . 129

VII. Interconnect Corners Considering Crosstalk Noise . . . 131

7.1 Problem Description . . . 133
7.1.1 Non-Monotonic Victim-Stage Delay . . . 136
7.1.2 Dependence on Coupling Capacitance . . . 137
7.1.3 Correlations between Coupling and Ground Capacitance . . . 140
7.2 Proposed Approach . . . 141
7.3 Experimental Results . . . 145
7.3.1 Results on Benchmark Circuits . . . 148
7.4 Summary . . . 150

VIII. Conclusion and Future Work . . . 151

APPENDICES **153**

BIBLIOGRAPHY **159**

LIST OF FIGURES

Figure

1.1 Timing graph of an example circuit along with the timing window at sink node of the circuit. . . . 3

1.2 Shrinking of wire geometries in the nanometer process technology leads to an increase in the amount of coupling capacitance. . . . 4

1.3 The aggressor transition results in the coupling-noise pulse on the victim due to the current flowing through the coupling capacitance. . . . 5

1.4 Delay noise due to the simultaneous switching of the aggressor-victim nets. . . . 6

1.5 Temporal filtering of aggressors using the criteria of overlap between the aggressor-victim timing windows. . . . 9

1.6 Decoupling of the aggressor-victim nets by using Miller-coupling capacitance. . . . 10

1.7 The actual victim waveform is obtained by linear superposition of the noiseless waveform and the coupling-noise pulse. . . . 13

1.8 Aggressor timing window and resulting noise envelope. . . . 14

1.9 Worst-case delay noise on a victim, coupled with two aggressors, using the cumulative-noise envelope. . . . 15

1.10 Nonlinear dependence of coupling-noise pulse on the aggressor alignment. . . . 16

1.11 Percentage increase in the circuit delays due to coupling noise. . . . 17

1.12 Percentage increase in circuit delays of noise-optimized industrial circuits due to coupling noise. . . . 18

1.13 Trends of the interconnect spacing and RC interconnect delay of global interconnects in future technology nodes. . . . 19

2.1 Victim alignment for worst-case delay, and the possible crossover of the noisy victim-output transitions. . . . 24

2.2 Stage delay and output arrival time of a victim as a function of the skew between aggressor-victim input transitions. . . . 26

2.3 Capacitively coupled aggressor-victim nets. . . . 32

2.4 Late victim-output transition crossing an early victim-output transition. 38

2.5 Maximize the victim receiver-output arrival time. 44

2.6 Victim net coupled to multiple aggressors. 46

2.7 HSPICE simulation plots showing the aggressor-victim output transitions. 49

3.1 Aggressor alignment maximizes the victim-stage delay but not the victim receiver-output arrival time. 55

3.2 Waveforms at the input of the victim-receiver gate. 57

3.3 Proposed methodology of performing noise analysis. 59

3.4 Cumulative Gate Overdrive Voltage ($CGOV$). 61

3.5 Scatter plot of $CGOV$. 63

3.6 Worst-case aggressor alignment for MIN analysis. 65

3.7 The experimental aggressor-victim circuit. 69

3.8 Histogram of %Error in delay noise for MAX analysis. 70

3.9 Histogram of %Error in delay noise for MIN analysis. 71

4.1 Indirect aggressors a2, a3 affect the timing window of the primary aggressor a1 in the noise analysis for victim v1. 74

4.2 Non-monotonicity in aggressor lists for aggressors a1, a2 and a3. 78

4.3 Pseudo-input noise envelope. 82

4.4 Dominance between the noise envelopes. 83

4.5 Partial ordering between noise envelopes. 86

4.6 Plot of circuit delay versus k for circuit i10. 90

5.1 Delay change curve captures the dependence of delay noise on input skew. 98

5.2 Skew window obtained by subtracting the early and late aggressor-input arrival times from the late victim-input arrival time. 99

5.3 Delay change curve in piece-wise quadratic form. 106

5.4 Comparison of analytical delay-noise distribution with that obtained from Monte-Carlo simulations. 109

5.5 Percentage reduction in circuit delay noise with statistical noise analysis. 111

6.1 Example of a signal path with coupled aggressors. 114

6.2 Monte-Carlo simulations to obtain the delay-noise distribution of a signal path. . . 117

6.3 In worst-case noise analysis, delay noise is obtained by the linear superposition of the largest cumulative coupling-noise pulse, obtained with no restrictions on the alignment of aggressors, with the noiseless-victim transition. 122

6.4 Perform worst-case noise analysis on the top-k aggressors and statistical noise analysis on the rest of the aggressors. 126

6.5 Histogram of the pessimism reduction in the delay noise of the top-400 critical paths for ckt1 (Note: unequal X-axis scales). 129

7.1 A cross-section of interconnect metal layers. 134

7.2 Elmore delay versus HSPICE-simulated delay as a function of the interconnect width (W). 137

7.3 Switching of aggressor-victim nets in the same direction. 138

7.4 Dependence of the victim-stage delay on the coupling capacitance. 139

7.5 Victim-stage analysis with Miller-coupling capacitance. 143

7.6 Victim-stage delay obtained at different interconnect corners. 146

7.7 Victim-stage delay at the proposed interconnect corner versus golden corner. . . . 148

LIST OF TABLES

Table

2.1 Results for the proposed latest victim-alignment approach. 50

3.1 Parameters of the experimental aggressor-victim circuit. 68

3.2 Mean %Error in delay noise of the victim compared to golden analysis. 72

4.1 Creating higher-order irredundant sets. 87

4.2 Validation of the proposed approach with brute-force enumeration for circuit i1. . . 89

4.3 Delay and runtime results for computing the top-k aggressors' addition set. 90

4.4 Delay and runtime results for computing the top-k aggressors' elimination set. . . . 91

5.1 Pessimism reduction in the circuit delay with statistical noise analysis. 110

6.1 Pessimism reduction in path delay noise by performing statistical noise analysis. . 127

6.2 Reduction in the total number of critical paths for ckt5 with statistical noise analysis. 128

7.1 Fast-path delays for LGSynth91 benchmark circuits. 149

LIST OF APPENDICES

Appendix

A. Inequality Between Victim Load Currents 154

B. First and Second Moments of Delay-Noise Distribution 157

ABSTRACT

Scaling of device dimensions into the nanometer process technology has led to a considerable reduction in the gate delays. However, interconnect delays have not scaled in proportion to gate delays, and global-interconnect delays account for a major portion of the total circuit delay. Also, due to process-technology scaling, the spacing between adjacent interconnect wires keeps shrinking, which leads to an increase in the amount of coupling capacitance between interconnect wires. Hence, coupling noise has become an important issue which must be modeled while performing timing verification for VLSI chips.

As delay noise strongly depends on the skew between aggressor-victim input transitions, it is not possible to a priori identify the victim-input transition that results in the worst-case delay noise. This thesis presents an analytical result that would obviate the need to search for the worst-case victim-input transition and simplify the aggressor-victim alignment problem significantly. We also propose a heuristic approach to compute the worst-case aggressor alignment that maximizes the victim receiver-output arrival time with current-source driver models. We develop algorithms to compute the set of *top-k* aggressors in the circuit, which could be fixed to reduce the delay noise of the circuit. Process variations cause variability in the aggressor-victim alignment which leads to variability in the delay noise. This variability is modeled by deriving closed-form expressions of the mean, the standard deviation and the correlations of the delay-noise distribution. We also propose an

approach to estimate the confidence bounds on the path delay-noise distribution. Finally, we show that the interconnect corners obtained without incorporating the effects of coupling noise could lead to significant errors, and propose an approach to compute the interconnect corners considering the impact of coupling noise.

CHAPTER I

Introduction

The tremendous advancements in semiconductor industry over the last few decades can largely be credited to the aggressive scaling of process technology. The devices and their interconnects are rapidly being scaled down in size and are being packed with increasing densities on Very Large Scale Integration (VLSI) chips. Today, it is common for VLSI chips to have transistors with feature sizes of 45 nanometers and the feature sizes expected to shrink further in future process technologies. However, the continuous scaling of device dimensions has led to several key challenges in timing verification of VLSI chips. With the aggressive scaling of interconnect wires, signal-integrity issues such as crosstalk noise have become important and must be addressed. Hence, it has become imperative to model the effects of crosstalk noise while performing timing verification of VLSI chips in the nanometer process technology.

1.1 Static Timing Analysis

The performance of VLSI chips is typically characterized by the operating clock frequency. In every clock cycle, each chip may be subjected to different operating conditions such as different input vectors, temperature and voltage profiles. Timing analysis is a methodology used by circuit designers to verify whether a chip can meet the target operating frequency. The goal is to verify that the signals at the output of

the circuit arrive at the right time (i.e., neither too early, nor too late) and thereby guarantee proper circuit operation. Therefore, for a given circuit topology, timing analysis propagates the signals to the outputs of the circuit and verifies whether all timing constraints are satisfied.

Static timing analysis (STA) is a widely adopted approach to perform timing verification. Unlike vector-based timing-simulation approaches, STA does not require an exhaustive simulation of all the critical paths in the circuit. Instead, STA propagates the signal delays to circuit outputs using the Critical Path Method (CPM) which is a widely-used technique in project planning and management. The term *static* refers to the fact that in STA, timing verification is performed in a manner that is independent of the *dynamic* input vectors. Instead, the worst-case delay of the circuit is computed, over all possible input vector combinations, by performing a single traversal of the circuit graph. Therefore, STA has a runtime that is linear with respect to the circuit size and can be used to efficiently perform timing verification of very large circuits. Additionally, STA results in a pessimistic analysis of the circuit delay, since it overestimates the delays of the critical paths in the circuit. This conservatism introduced into the timing verification by STA provides additional guard-banding, and makes timing verification robust in the face of modeling errors. Therefore, it is hardly surprising that STA has become the mainstay tool while performing timing verification of VLSI chips over the last few decades.

An example circuit and its associated timing graph is shown in Figure 1.1 where the vertices consist of logic gates in the circuit along with primary inputs and outputs of the circuit. A *critical path* is defined as the signal path between an input pin and an output pin of the circuit having the maximum delay. The delay of the critical path determines the operating frequency of the circuit. The *arrival time* of a signal

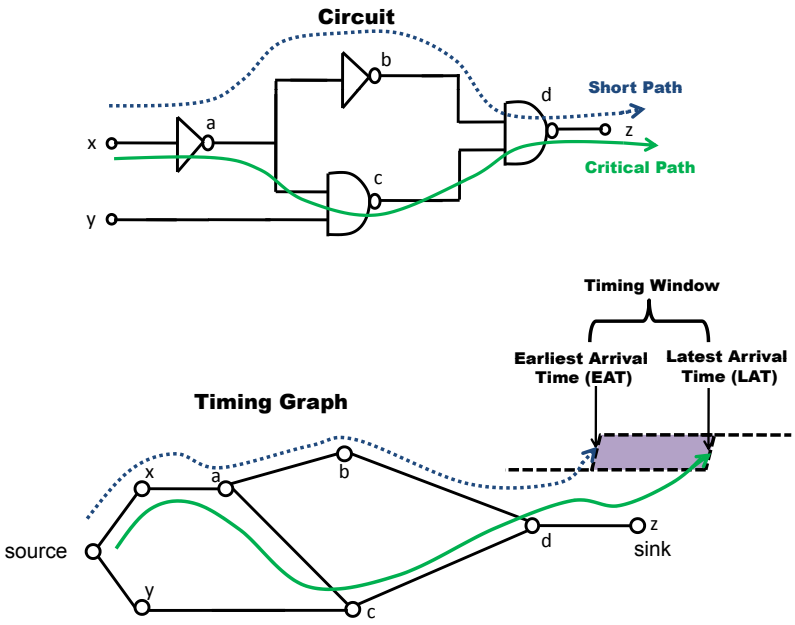


Figure 1.1: Timing graph of an example circuit along with the timing window at sink node of the circuit.

is the amount of time elapsed before the signal arrives at a certain point in the circuit as it propagates from the primary inputs of the circuit. The signal arrival time is obtained by summing up the delays of all the gates and interconnects present in the signal path. In STA, arrival times are often represented by the pair, the *earliest-arrival time* (EAT) and the *latest-arrival time* (LAT), where EAT refers to the earliest possible time at which a signal can change, and similarly, LAT refers to the latest possible time at which the signal can change. A *timing windows* is defined as the time interval between EAT and LAT within which the signals are allowed to change (as shown in Figure 1.1). It follows from the definition of a critical path that the output node of the critical path has the maximum LAT.

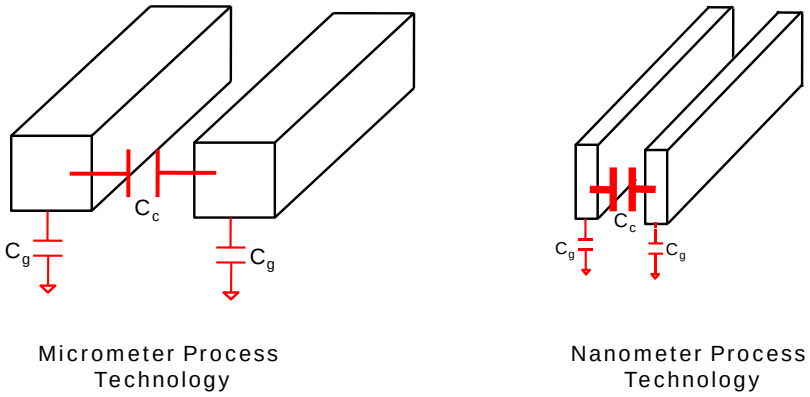


Figure 1.2: Shrinking of wire geometries in the nanometer process technology leads to an increase in the amount of coupling capacitance.

1.2 Crosstalk Noise

As the feature sizes have been shrinking with process-technology scaling, the spacing between adjacent interconnect wires keeps decreasing in every process technology. Also, while the lateral width of interconnect wires has been scaled down significantly their vertical height has not been scaled in proportion (as shown in Figure 1.2). Both these trends lead to a very rapid increase in the amount of coupling capacitance (essentially like parallel-plate capacitors) between the wires. In [69], it was reported that coupling capacitance accounts for more than 85% of the total interconnect capacitance in the $90nm$ technology node. More aggressive technology scaling will only lead to an increase in the overall contribution of the coupling capacitances to the total interconnect capacitance. Therefore, signal-integrity issues such as crosstalk noise have become important when performing timing verification of VLSI chips.

Due to capacitive coupling, the switching characteristics of a net is affected by simultaneous switching of nets that are in close physical proximity. The net under analysis which suffers from coupling noise is referred to as *victim*, and all neighboring nets which contribute to coupling noise on the victim are termed as *aggressors*. Figure

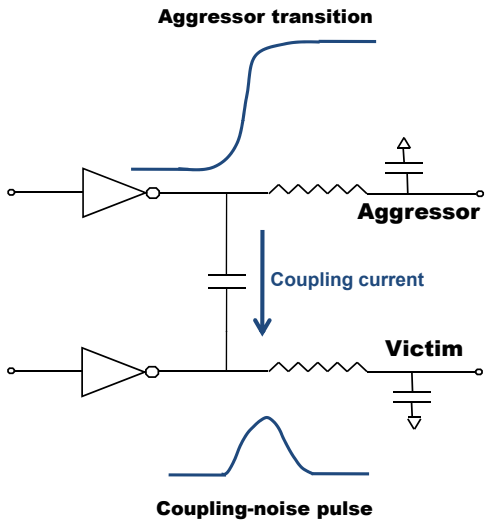


Figure 1.3: The aggressor transition results in the coupling-noise pulse on the victim due to the current flowing through the coupling capacitance.

1.3 illustrates coupling noise injected on the victim due to the rising transition of its aggressor. As the aggressor transition occurs, the voltage at the victim gets pulled up due to the AC current flowing through the coupling capacitance. The resulting glitch in the victim voltage due to the aggressor transition is referred to as *coupling-noise pulse*. The peak of the coupling-noise pulse usually occurs when the aggressor transition is completed and there is no more coupling current flowing through the coupling capacitance. Finally, the coupling-noise pulse gradually dies down once the aggressor transition is completed and the victim node discharges the accumulated charge. It has become imperative to model the signal-integrity issues that can arise due to the charge transfer through the coupling capacitances for VLSI chips in the nanometer process technology.

Consider another scenario, where both aggressor-victim nets are switching at the same time. We seek to model the change in the switching characteristics of victim due to the coupling noise from the aggressor. If the aggressor-victim pair switch in mutually opposite directions, then the coupling-noise pulse can slow down the

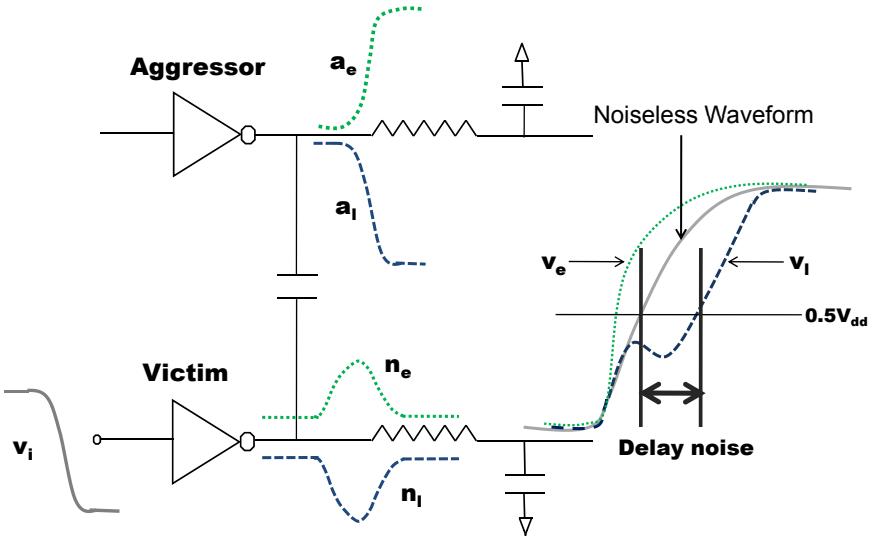


Figure 1.4: Delay noise due to the simultaneous switching of the aggressor-victim nets.

victim transition (e.g., v_l in Figure 1.4) and increase the victim arrival time. On the other hand, if the aggressor-victim pair switch in the same direction, then the coupling-noise pulse can speedup the victim transition (e.g., v_e in Figure 1.4) and reduce the victim arrival time. This change in the victim arrival time due to coupling noise is referred to as *delay noise*. Delay noise contributes to a significant portion of the circuit delay for high performance VLSI circuits. Furthermore, as technology advances, we see an increasing chip frequency and decreasing voltage margin. All of the above trends exacerbate the impact of crosstalk noise on the victim-stage delay. Therefore, it has become necessary to accurately model the delay noise while performing timing analysis for VLSI chips in the nanometer process technology.

1.3 Crosstalk-Noise Filtering

In practice, a typical victim net is capacitively coupled to numerous aggressor nets located spatially adjacent to the victim. Therefore, performing noise analysis

on the victim by accounting for crosstalk noise from all the aggressors would be very expensive computationally. Typically, one would like to analyze all the aggressors that contribute to a significant amount of coupling noise and ignore the rest. We can efficiently reduce the aggressor search space by exploiting the spatial, temporal and functional properties of the aggressor-victim network and excluding those aggressors that do not cause any significant coupling noise on the victim.

1.3.1 Spatial Filtering

We know that crosstalk noise occurs due to charge sharing through the coupling capacitances between the aggressor-victim nets. Hence, the amount of coupling noise depends strongly on the magnitude of the coupling capacitance. If two nets lie adjacent to each other in the layout, then there may be a significant amount of coupling capacitance between them. Conversely, for nets that are not located spatially adjacent to each other, the coupling capacitance could be insignificant. Hence, the coupling noise due to these aggressors would be insignificant and could be ignored while performing noise analysis on the victim. Therefore, one can use the capacitance-extraction tools to report only the significant aggressors for a victim and filter out the rest.

1.3.2 Functional Filtering

Traditionally, noise-analysis tools are conservative and they estimate the worst-case delay noise for every victim by assuming that all the aggressor transitions are mutually correlated. In order to compute the maximum slow-down due to coupling noise, all the aggressors are assumed to transition simultaneously in an opposite direction with respect to the victim transition. Similarly, in order to compute the maximum speed-up, all aggressors are assumed to switch simultaneously in the same

direction as the victim transition.

However, in any given clock cycle, the switching activity of each aggressor in the circuit is determined by the state of the input vectors and the circuit topology. Therefore, the assumption that all aggressors must transition in the same direction is often pessimistic. Logical constraints between the aggressors can be used to prune the set of aggressors and eliminate those that can never cause coupling noise because of their logical constraints. In [67], using Lagrangian Relaxation and network flow based approaches, the authors compute the subset of aggressors that maximizes the crosstalk induced delay noise on a coupled victim net under given logical constraints.

1.3.3 Temporal Filtering

We know that delay noise on a victim can occur only when the victim and aggressor transitions occur in close temporal proximity of each other. As defined earlier, the timing window of a net identifies an interval in the clock period within which the net can transition. Hence, we can use the information from the timing windows of the aggressor-victim nets to determine whether both can switch at the same time. A simple heuristic often used to determine whether the aggressor can couple noise on the victim is to check whether the respective aggressor-victim timing windows overlap in time (e.g., aggressors a2 and a3 in Figure 1.5). Now, both aggressor-victim nets can switch in the overlap region of the aggressor-victim timing windows. Consequently, we can filter out those aggressors whose timing windows do not overlap with the victim timing window (e.g., aggressor a1 in Figure 1.5), since the aggressor-victim transitions cannot occur at the same time.

It must be noted that the computation of delay noise and timing windows are mutually dependent. Delay noise cannot be computed accurately before the timing windows are defined. Conversely, accurate timing windows cannot be computed

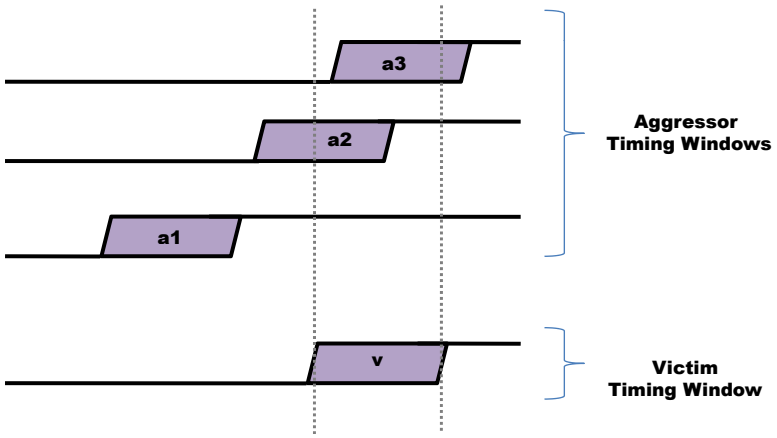


Figure 1.5: Temporal filtering of aggressors using the criteria of overlap between the aggressor-victim timing windows.

without any information about the delay noise. In [61, 12, 77], it was shown that this *chicken-and-egg* problem can be solved using an iterative approach. The iterations start with either the assumption that all aggressor timing windows overlap with that of the victim, or that there is no overlap between the aggressor-victim timing windows. In each iteration, the worst-case aggressor-victim alignment is determined by updating the timing windows with delay noise computed in the previous iteration. Delay noise is then recomputed and timing windows are updated accordingly until the two converge. It was shown in [61] that this iterative method is guaranteed to converge, and its convergence was theoretically established in [77].

1.4 Miller-Coupling Capacitance

A simple way of performing timing analysis in the presence of crosstalk noise is to replace the coupling capacitance by an equivalent Miller capacitance to account for the slow down or the speed up of the victim transition. Hence, we are effectively modeling the delay noise on the victim by simply scaling the coupling capacitance with the appropriate *Miller-coupling factor* (MCF). The simple approximation leads

to a decoupling between the aggressor-victim interconnects (as shown in Figure 1.6) and allows us to compute the signal arrival times using the existing STA framework.

The exact value of the Miller-coupling factors for the victim (MCF_{victim}) depends on the mutual switching directions and the ratio of the slopes of the aggressor-victim waveforms

$$MCF_{victim} = 1 \pm \frac{\Delta a}{\Delta v}. \tag{1.1}$$

If the aggressor-victim nets switch in mutually opposite directions (as shown in Figure 1.6), then MCF_{victim} is obtained by adding 1 to the relative ratio of the slopes ($\frac{\Delta a}{\Delta v}$). Conversely, if the aggressor-victim nets switch in the same direction, then MCF_{victim} is obtained by subtracting $\frac{\Delta a}{\Delta v}$ from one. Suppose the aggressor-victim nets switch in the opposite directions with equal slopes (i.e., $\Delta a = \Delta v$). Using

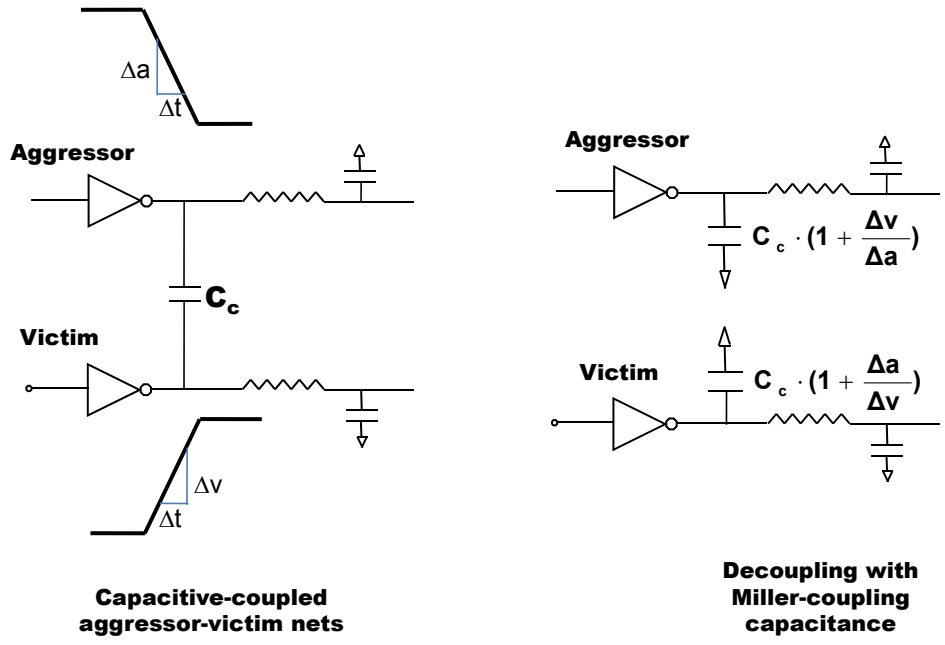


Figure 1.6: Decoupling of the aggressor-victim nets by using Miller-coupling capacitance.

Equation 1.1, we can obtain the commonly used MCF_{victim} value of two. In such cases, the Miller capacitance used to decouple the aggressor-victim interconnects is twice the size of the original coupling capacitance. Similarly, if the aggressor-victim nets switch in the same direction with equal slopes, then we obtain an MCF_{victim} value of zero. However, the above estimates are not always conservative, and in [46] the authors show that the MCF_{victim} could lie within the range [-1, 3]. Timing verification can be performed by using the existing STA framework [61, 22, 73] by using the MCF_{victim} to appropriately scale the coupling capacitance for each victim.

1.5 Crosstalk-Noise Analysis

Delay noise on the victim depends on numerous factors such as the aggressor-victim alignment, slew rates, and drive strengths of the aggressor-victim pair. Often, noise analysis with Miller-coupling capacitances does not provide adequate accuracy. Hence, although the use of Miller-coupling capacitance is computationally very efficient, they provide only a first-order approximation of the coupling noise and are not very accurate. Therefore, alternative approaches have been proposed to estimate the delay noise on the victim more accurately. We will first briefly review the framework for performing noise analysis with linear-driver models, and then look at the issues that must be addressed while modeling the nonlinearity of the aggressor-victim drivers.

1.5.1 Noise Analysis with Linear Drivers

Logic gates have delays that are inherently nonlinear functions of circuit parameters such as the output capacitance and the transition time of the input signals. Hence, we must essentially solve a nonlinear problem to accurately compute the delay noise on the victim. However, nonlinear simulations of aggressor-victim drivers

and interconnects can be computationally very expensive. Hence, linear-driver models of the gates are typically constructed and used for performing efficient timing and noise analysis, since they allow the use of the principle of linear superposition. A logic gates is modeled by its Thevenin model which is composed of a ramp-voltage source and a series output resistance. Using the linear-superposition assumption, the *noisy*-victim waveform can be approximated by the linear sum of the coupling-noise pulse and the noiseless-victim waveform.

The coupling-noise pulse is first computed by replacing the aggressor driver with its Thevenin model and the victim driver with its Thevenin resistance (as shown in Figure 1.7(a)). Next, the noiseless-victim waveform is obtained by modeling the victim driver with its Thevenin model and the aggressor with its Thevenin resistance (as shown in Figure 1.7(b)). Finally, the voltage waveforms obtained from these two simulations are combined together, using the linear-superposition assumption, to obtain the noisy-victim waveform (as shown in 1.7(c)). It is trivial to extend the noise analysis for the case when the victim is coupled to several aggressors by using the linear-superposition assumption and combining the coupling-noise pulses injected from all the aggressors.

In noise analysis, an important issue that must be determined is the relative alignment of the coupling-noise pulse with respect to the noiseless-victim transition. In worst-case noise analysis, we must search for an aggressor transition (referred to as the worst-case aggressor alignment) which results in the coupling-noise pulse that maximizes the delay noise when superimposed with the noiseless-victim transition. As mentioned earlier, a timing window identifies an interval in the clock period within which a net can transition. The early-arrival time (EAT) and the late-arrival time (LAT) is computed by propagating the fastest and the slowest switching signals from

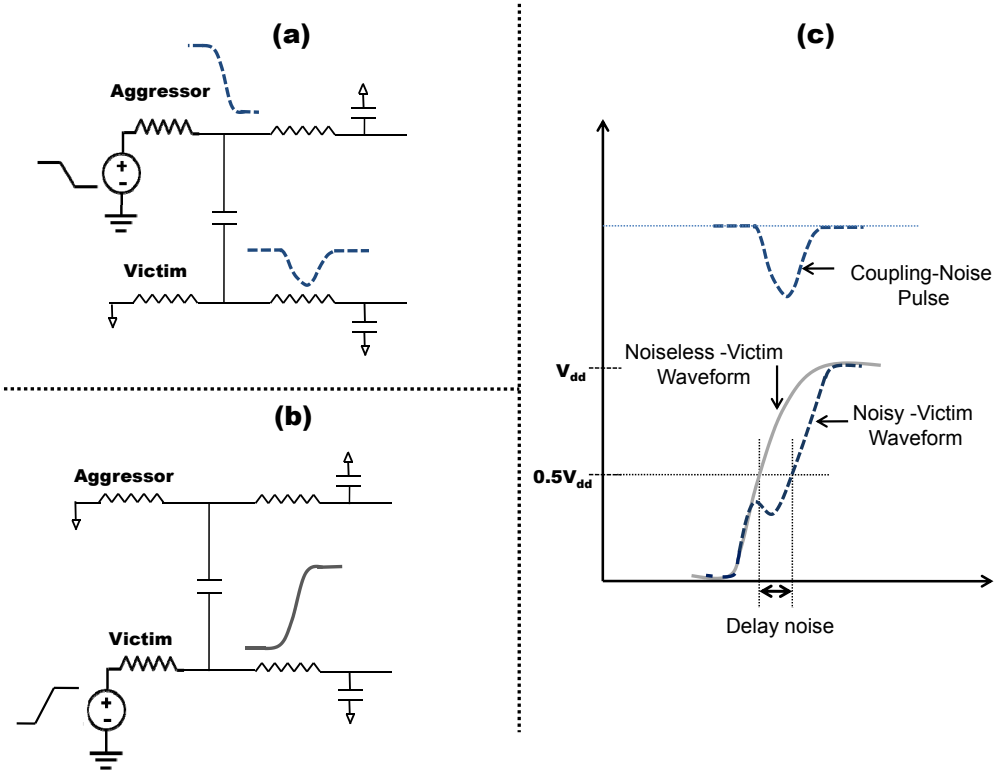


Figure 1.7: The actual victim waveform is obtained by linear superposition of the noiseless waveform and the coupling-noise pulse.

the primary inputs of the circuit. Hence, any feasible aggressor transition, including the worst-case aggressor transition, must lie within the boundaries of the timing window of the corresponding aggressor net.

The problem of computing the worst-case aggressor alignment is simplified under the linear-superposition assumption. Using coupling-noise pulses and the information about aggressor timing windows, one can compute a *noise envelope* that bounds the noise coupled from an aggressor to the victim net. As shown in Figure 1.8, the noise envelope for a particular aggressor is obtained by sweeping the aggressor transition within its timing window and tracking the peak of the coupling-noise pulse injected on the victim net. The *trapezoidal*-shaped noise envelope can be computed efficiently by combining the coupling-noise pulses coupled from an aggressor transition occurring

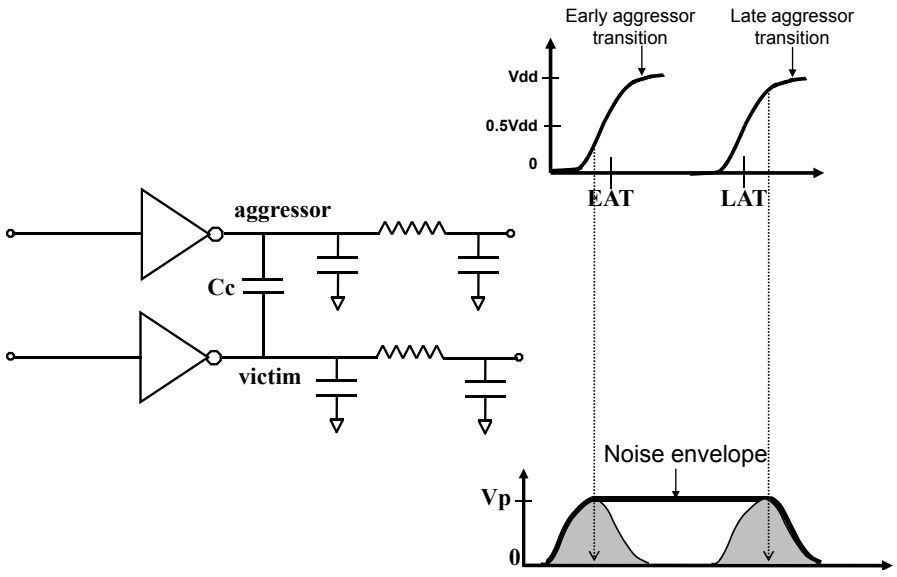


Figure 1.8: Aggressor timing window and resulting noise envelope.

at its EAT and its LAT and subsequently connecting their noise peaks (as shown in Figure 1.8).

The worst-case delay noise due to an aggressor is obtained by superimposing the corresponding noise envelope with the noiseless-victim waveform and observing the change in the victim delay. If multiple aggressors are coupled to the victim, then we construct noise envelopes for each aggressor separately, and combine them together to create a cumulative-noise envelope (as shown in Figure 1.9). The worst-case delay noise due to all aggressors is obtained by superimposing the cumulative-noise envelope with the noiseless-victim waveform.

1.5.2 Noise Analysis with Nonlinear Drivers

Industrial noise-analysis tools [49] use linear-driver models for performing very fast and efficient noise analysis. However, the victim driver-output resistance exhibits a nonlinear behavior during the time period when the victim driver-output transition occurs. Hence, assuming a fixed output resistance for the victim driver can lead to errors in the computation of the coupling-noise pulse. We demonstrate, by plotting

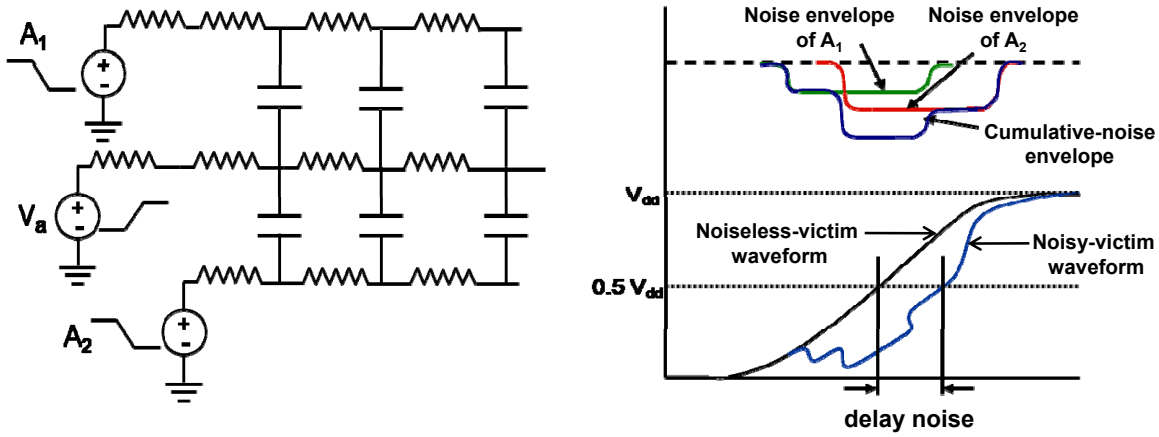


Figure 1.9: Worst-case delay noise on a victim, coupled with two aggressors, using the cumulative-noise envelope.

in Figure 1.10, the coupling-noise pulses obtained by sweeping the alignment of the aggressor relative to the victim transition. The coupling-noise pulse was generated for every aggressor alignment by subtracting the noisy-victim waveform from the noiseless-victim waveform. Figure 1.10 illustrates that coupling-noise pulse changes in size as the aggressor alignment is varied. In fact, the coupling-noise pulse is largest when both aggressor-victim nets are switching at the same time. Therefore, the principle of superposition is not applicable for nonlinear drivers as the peak of the coupling-noise pulse is not constant and instead changes with the aggressor alignment.

In order to model the nonlinear behavior of the coupling-noise pulse with respect to the aggressor alignment, we must model the nonlinear shape of the noise envelope as shown in Figure 1.10. This requires multiple coupling-noise simulations with different aggressor alignments, and can be very expensive computationally. In [15], it was shown that the worst-case aggressor alignment is a nonlinear function of the victim slew, the coupling-noise pulse, and the victim receiver-output load. The authors proposed the use of precharacterized look-up tables to identify the worst-

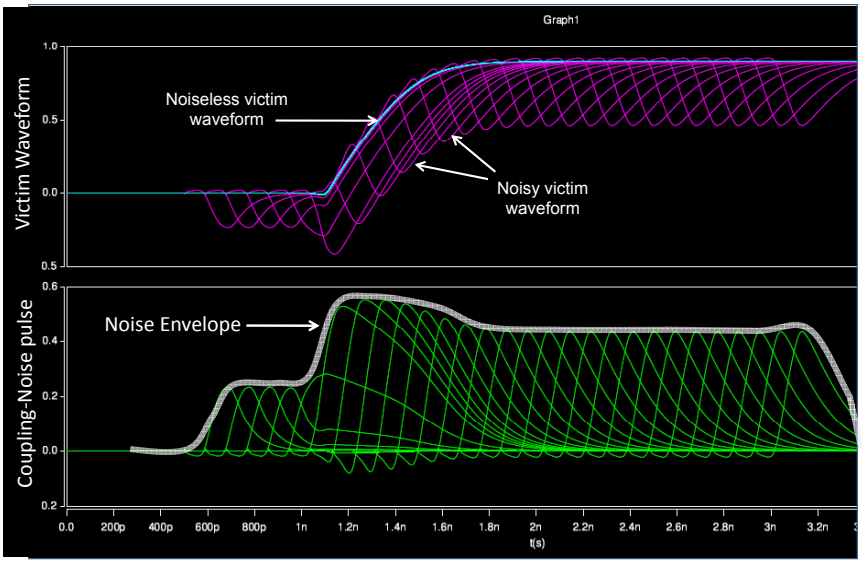


Figure 1.10: Nonlinear dependence of coupling-noise pulse on the aggressor alignment.

case aggressor alignment. In [47], the worst-case aggressor alignment was computed using the techniques of constrained optimization with the objective of maximizing the victim delay noise.

1.5.3 Impact of Coupling Noise on Design Performance

The amount of coupling noise in a design depends on the amount of wiring congestion that is present in the design. Hence, for circuits with a lot of wiring congestion, coupling noise can cause a significant degradation in the circuit performance. Therefore, worst-case noise analysis must always be performed while doing the timing verification of VLSI chips at every stage of the design cycle. Noise analysis, when performed early in the design cycle such as after initial place-and-route, allows us to catch potential congestion problems that can occur later, and helps to speed up the design turnaround time. More accurate noise analysis can also be performed during signoff, for example, after buffer insertion and noise-aware global routing when we have accurate post-layout interconnect coupling capacitances data. Hence, circuit designers can use noise-analysis tools, at every stage of the design cycle, to flag

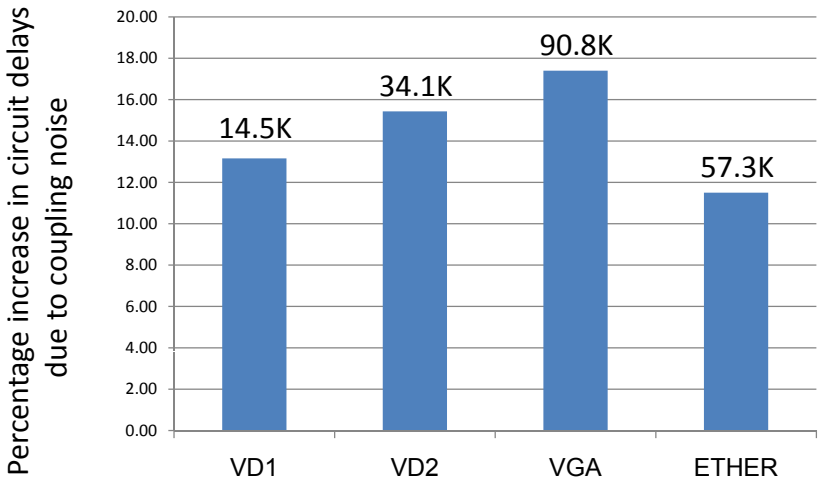


Figure 1.11: Percentage increase in the circuit delays due to coupling noise.

potential violations and ensure the reliable operation of VLSI chips throughout its lifetime.

In this subsection, we demonstrate the impact of coupling noise on the performance of VLSI chips. The experiments were performed on the following circuits, Viterbi Decoder 1 (VD1), Viterbi Decoder 2 (VD2), Ethernet MAC Core (ETHER), and VGA Controller Core (VGA), with the circuit sizes varying between 15,000 to 90,000 gates. Up to *seven* metal layers were used to perform the place-and-route of the circuits with a target floorplan utilization of 60%. The Cadence Encounter tool was used to perform the place-and-route of the circuits in the 65nm process technology. The same tool was used to extract the post-layout parasitic coupling capacitances of the interconnects. A prototype delay-noise analysis tool was implemented in C++, which uses the extracted coupling capacitances data and performs delay-noise analysis on the circuits. For these circuits, we see in Figure 1.11 that the coupling noise contributes between 11.5%-17.4% to the circuit delays.

Since, coupling capacitances depend on the amount of wiring congestion that is present in a design, reducing the wiring congestion in a design leads to a reduction

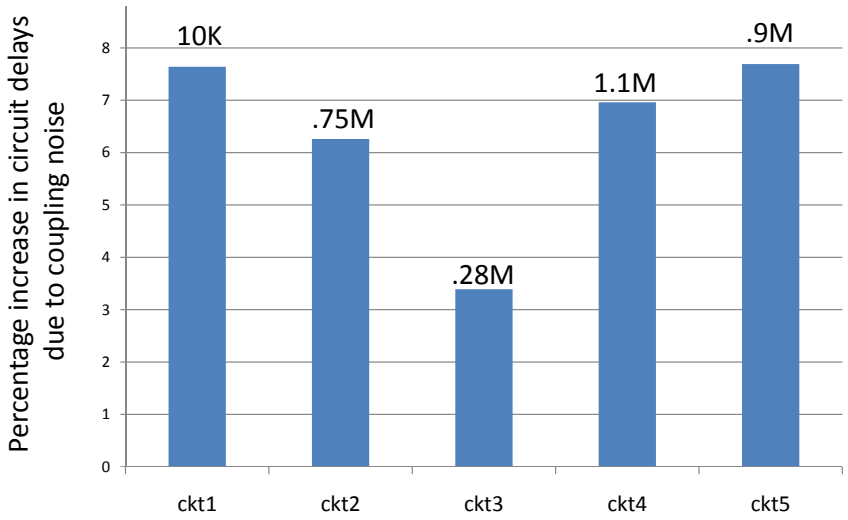


Figure 1.12: Percentage increase in circuit delays of noise-optimized industrial circuits due to coupling noise.

in the amount of the parasitic coupling capacitances, and consequently results in a reduction in the amount of coupling noise in the design. The commonly-used techniques for coupling-noise mitigation include sizing up the victim driver, increasing the aggressor-victim interconnect spacing, shielding the victim interconnects, routing of the aggressor-victim interconnects in different metal layers, and inserting buffers to reduce the temporal overlap between the aggressor-victim nets, etc.

Industrial designs are often optimized such that they are clean from any major noise violations before they are ready for tape-out. For such carefully designed circuits, coupling noise may have a reduced impact on the delays of critical paths. Figure 1.12 shows the impact of delay noise on the performance of real industrial circuits having millions of instances in the 65nm process technology. The designs were cleaned of any major coupling-noise violations by using the noise-mitigation techniques detailed above. Noise analysis was performed with an industrial noise-analysis tool (Primetime-SI) using the detailed interconnect capacitances which were extracted post layout. For these noise-optimized industrial circuits, we see in Figure

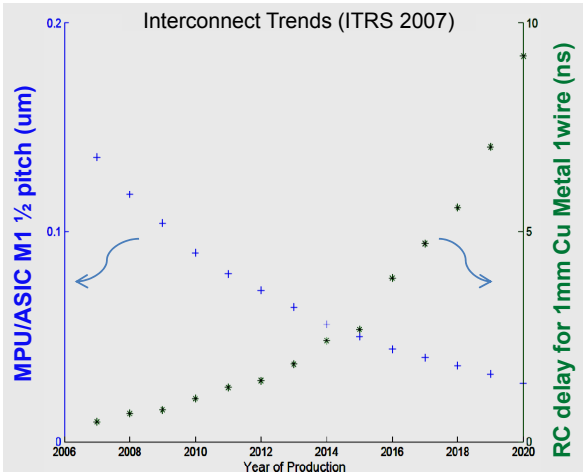


Figure 1.13: Trends of the interconnect spacing and RC interconnect delay of global interconnects in future technology nodes.

1.12 that coupling noise contributes between 3.39%-7.64% to the signal path delays.

Today, we see increasing chip frequencies and decreasing voltage margins. Also, with process-technology scaling, the device dimensions and the wire spacings are reducing rapidly. In Figure 1.13, we plot the future trends of the half-pitch spacing between interconnect wires. The spacing between interconnect wires is expected to reduce significantly in future technology nodes, which will lead to an increase in the amount of coupling capacitances of interconnect wires. We also plot the future trends of the RC interconnect delay for a 1mm long global interconnect wire. The trends indicate a significant increase in the global interconnect delay at future process-technology nodes, and the global interconnect delay will contribute significantly to the total circuit delay. Hence, coupling-noise issues will be significant in future process-technology nodes, especially for global interconnects and clock networks, where it is common practice to use long interconnect wires. For such wires, coupling noise can contribute to a significant part of the interconnect delay. Consequently, in future process technologies, we expect to see a further increase in the amount of

coupling noise and its impact on the circuit delay. Therefore, it will be necessary to accurately model the delay noise while performing timing analysis for VLSI chips in the nanometer process technology.

1.6 Thesis Overview and Key Contributions

In this research, we focus on analysis and modeling of crosstalk noise for VLSI chips in the nanometer process technology. We propose solutions [31, 34, 32, 33, 35, 30] that aim at solving key problems associated with the accurate analysis and modeling of crosstalk-noise effects. The outline of the thesis is as follows:

- **Victim alignment in crosstalk-aware timing analysis**: As delay noise strongly depends on the relative alignment between the aggressor-victim transitions, it is not possible to a priori identify the victim transition that results in the worst-case delay noise. In Chapter 2, we present an analytical result that would obviate the need to search for an *optimal* victim transition in its timing window. Using the properties of standard nonlinear CMOS drivers, we show that the latest output arrival time of a victim net occurs only when its input transition occurs at the latest point in its timing window. The above result leads to a significant speed-up in the noise analysis without compromising accuracy.

- **Worst-case aggressor-victim alignment with current-source driver models**: In Chapter 3, we propose a heuristic approach to compute the worst-case aggressor alignment that maximizes the victim receiver-output arrival time with current-source driver models. A cumulative gate overdrive voltage (CGOV) metric is defined to model the total victim receiver-output current. The aggressor alignment that results in the minimum CGOV corresponds to the slowest victim receiver-output transition. Simulations performed on industrial nets to

validate the proposed methodology show an average error of 1.7% in delay noise compared to the worst-case alignment obtained by an exhaustive sweeping.

- **Top-k Aggressors in Noise Analysis**: With limited resources for fixing noise violations, circuit designers would like to minimize the total number of aggressors that need to be fixed. In Chapter 4, we present a novel technique for computing the set of top-k critical aggressors. It is shown that the computation of the set of the top-k aggressors is nontrivial, since we must consider all permutations of the aggressors coupled to a critical path. The proposed algorithm uses two novel concepts to provide a tractable solution: Firstly, we model the propagated delay noise with a pseudo-input noise envelope, which results in an efficient problem partition. Secondly, we prune the enumeration space by using the property of dominance among noise envelopes and imposing a partial ordering between aggressor sets. Finally, the effectiveness of our proposed algorithm is demonstrated on benchmark circuits.

- **Modeling crosstalk in statistical static timing analysis**: Process variations lead to variability in the aggressor-victim alignment which results in variability of the delay noise. In Chapter 5, we model the variability of delay noise by deriving closed-form expressions of the mean, the standard deviation and the correlations of the delay-noise distribution. The computed canonical delay-noise distribution can easily be integrated with existing statistical static timing analysis (SSTA) tools.

- **Pessimism reduction with path-based statistical noise analysis**: Worst-case noise analysis can be very pessimistic, and circuit designers would prefer less pessimistic noise-analysis techniques. In Chapter 6, we propose a statis-

tical framework for estimating the confidence bounds on the path delay-noise distribution. Experimental results on industrial circuits show that statistical noise analysis leads to an average reduction of 9.97% in the path delay noise compared to worst-case noise analysis.

- **Interconnect corners considering crosstalk noise**: Process variations have resulted in a significant variability in the interconnect delay. In Chapter 7, we show that the interconnect corners obtained assuming no coupling noise could be significantly different from the true interconnect corners. Therefore, it could lead to optimistic analysis, particularly for fast-path analysis performed to check hold-time violations. We propose an approach to compute the true interconnect corners considering coupling noise due to the simultaneous switching of aggressors.

- **Conclusions and future work**: In Chapter 8, we conclude the thesis, and discuss possible extensions to the solutions that were proposed in this thesis.

CHAPTER II

Victim Alignment in Crosstalk-Aware Timing Analysis

Delay noise is very sensitive to the relative alignment between the aggressor and victim transitions. Therefore, it is not trivial to find the worst-case alignment between the aggressor and the victim transitions, such that the output arrival time of the victim is maximized [40, 20]. For a better understanding of the problem, consider two transitions at the input of the victim, one switching earlier than the other (as shown in Figure 2.1). If the early victim-output transition couples more strongly with an aggressor or aligns with additional aggressors, then its coupling-noise pulse could be larger compared to that of the later victim-output transition. However, it is not clear whether a greater coupling-noise pulse in the earlier transition can result in a later victim-output arrival time. Hence, it is difficult to a priori determine which victim-input arrival time will produce the latest victim-output arrival time. To determine the worst-case alignment of the victim transition, we must compute the maximum delay noise for all possible victim transitions, and then pick the one that results in the latest output arrival time.

Initial approaches for performing noise analysis [61, 22, 46, 73] used Miller-coupling factors (e.g., 0-2) to appropriately scale the coupling capacitances. However, delay noise depends on numerous factors such as the aggressor-victim alignment, slew rates,

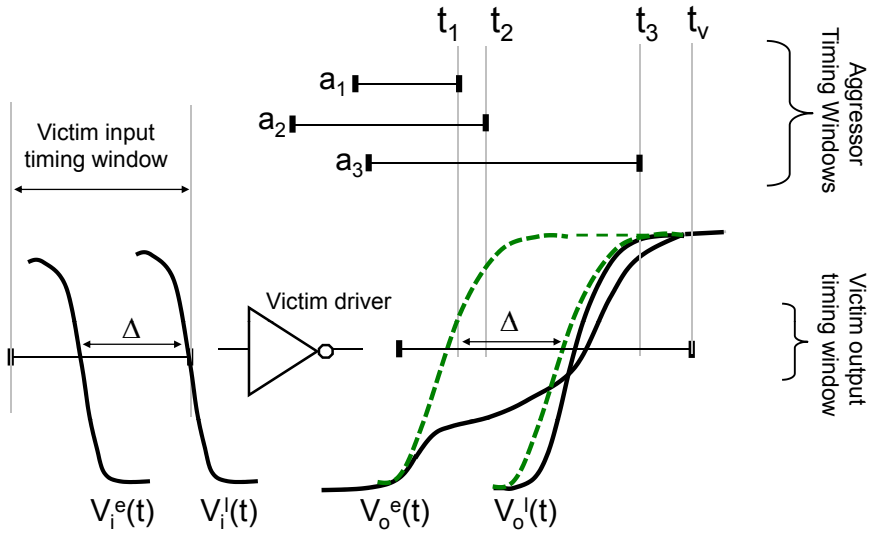


Figure 2.1: Victim alignment for worst-case delay, and the possible crossover of the noisy victim-output transitions.

and drive strengths of the aggressor-victim pair. Therefore, noise analysis performed by simply scaling the coupling capacitances does not provide adequate accuracy. In [62, 63], a relative window based approach is proposed where the delay noise is obtained as a function of the relative window for every aggressor-victim pair. The dependence of delay noise on the alignment can be computed by using SPICE based simulations [64] or derived analytically using curve-fitting techniques [7].

A brute-force solution to the aggressor-victim alignment problem can be obtained by sweeping the victim arrival time exhaustively within its timing window, finding the worst-case aggressor alignment for each victim transition and selecting the one that results in the latest victim-output arrival time. Since an exhaustive sweep of the victim-alignment is not practical for large circuits, several heuristic methods have been proposed [24, 40, 75, 13] to solve the victim-alignment problem. The authors formulate the alignment problem as a weighted-channel-density problem in [24], and empirical models are used to predict the alignment in [40]. An effective

skew-window model was proposed in [75] which leads to a pessimistic estimation of delay noise. In [13], the concept of effective delay noise was introduced to eliminate the pessimism by capturing the maximum change in the victim timing window due to coupling noise. Recently, in [47], the authors solve the alignment problem as a constrained-optimization problem by using nonlinear simulations for evaluating the nonlinear objective function.

All the approaches outlined above are either heuristics or perform computationally-expensive enumerations in the victim timing window to solve the victim-alignment problem. In contrast to these approaches, we present an analytical result that would obviate the need to enumerate the victim transition within in its timing window. Using the properties of standard nonlinear CMOS drivers, we show that the latest output arrival time of a victim net occurs only when its input transition is aligned at the latest point in its timing window. Since we only need to compute the worst-case alignment of the aggressors, the alignment problem is significantly simplified. This result has been empirically observed in the industry and is already used in industrial noise-analysis tools as an efficient heuristic to avoid enumerating the victim timing window. However, to the best of our knowledge, this is the first work which analytically shows that the above result holds for both linear and nonlinear driver models. Although, the proof is fairly straightforward for linear-driver models, it is nontrivial for nonlinear drivers.

Delay noise is a function of the aggressor-victim alignment, and delay noise can decrease when the victim-input arrival time is increased as shown in Region B of Figure 2.2. However, we analytically show that this decrease in delay noise is always less than the shift in the victim-input arrival time. In other words, the magnitude of the slope of the delay change curve in Region B is less than one (even for nonlin-

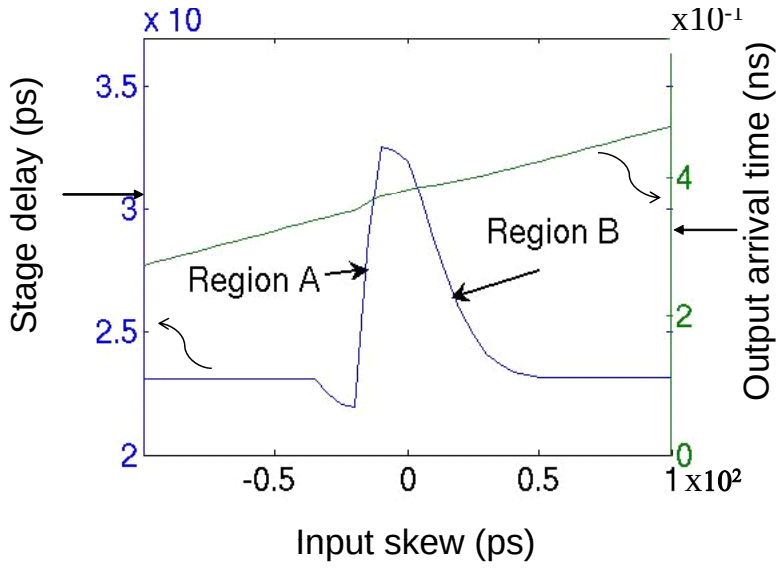


Figure 2.2: Stage delay and output arrival time of a victim as a function of the skew between aggressor-victim input transitions.

ear drivers). We also show that the victim-input transition must be aligned at its latest possible arrival time to maximize the victim-output arrival time. This result also reduces the complexity of the aggressor-victim alignment problem significantly, since we no longer need to search for an optimal victim transition within its timing window. Furthermore, the total number of aggressors which couple noise to the latest-occurring victim transition is always less than the number of aggressors that are coupled to a victim transition that can occur at any point in its timing window. Using the above approach, we obtain a significant speed-up over existing approaches without compromising accuracy.

2.1 Problem Description

The focus of this chapter is to find an optimal alignment of the victim transition for noise analysis. In this work, we analyze the case when the victim and the aggressor are switching in mutually opposite directions. In particular, we want to solve for a victim-*input* arrival time such that the delay noise results in the *latest* victim-output

arrival time. A similar analysis can be performed to find the earliest victim-output arrival time when the aggressor-victim nets are switching in the same direction.

Figure 2.2 illustrates the plot of the victim-stage delay as a function of the difference between the aggressor-victim input arrival times (referred to as the input skew). For large values of both positive and negative input-skew, there is no temporal overlap between the aggressor-victim transitions. Therefore, the victim-stage delay remains unchanged and is given by the nominal delay. However, for smaller values of input skew, which occurs when the aggressor-victim nets switch in close temporal proximity, the aggressor transition couples noise on the victim transition and affects its stage delay. As a thought experiment, suppose we have a fixed aggressor-input transition, and we can vary the arrival time of the victim-input transition. In Region A of the plot, the victim-stage delay increases with an increase in its input arrival time because the temporal overlap between the aggressor-victim transitions increase as the victim transition is further delayed in time. Once the victim and the aggressor transitions are optimally aligned and the victim-stage delay peaks, any further increase in the victim-input arrival time leads to misalignment. Consequently, it leads to a decrease in the stage delay (in Region B) due to a reduction in the amount of delay noise.

Suppose that the magnitude of the slope in Region B is always less than one. In other words, the decrease in the stage delay in Region B is always less than the increase in its input arrival time. Since the output arrival time is the sum of the input arrival time and the stage delay, the victim-output arrival time will always be a monotonic increasing function of the victim-input arrival time. Therefore, the latest victim-output arrival time will occur only when the victim-input transition occurs at the latest possible time, i.e., the latest point in its timing window. However, it is

nontrivial to show that the magnitude of the slope in Region B is always less than one especially for nonlinear drivers, since the analysis is complicated by the cyclic nonlinear dependence between the aggressor and the victim responses [47].

For example, consider the aggressors a_{1-3} coupled to a victim net and their respective timing windows (as shown in Figure 2.1). Let $v_i^e(t)$ and $v_i^l(t)$ be the early-late victim-input transitions, respectively, where $v_i^e(t)$ switches earlier in time than $v_i^l(t)$ and are separated in time by an amount $\triangle$. From causality arguments, the *noiseless* victim-output transitions (dashed waveforms) must also be separated by $\triangle$. It can be seen that the noisy victim-output transition $v_o^e(t)$, corresponding to the early victim-input transition, overlaps with the timing windows of all three aggressors a_{1-3}. However, when the victim-input transition is delayed, the resulting output transition $v_o^l(t)$ can only overlap with the timing window of aggressor a_3. Consequently, the delay noise observed for $v_o^e(t)$ is greater than that of $v_o^l(t)$. In such a case, if the difference between the delay noise of $v_o^e(t)$ and $v_o^l(t)$ is greater than $\triangle$, then the victim-output waveforms will cross each other (as shown in Figure 2.1). Therefore, for cases when there are multiple aggressors coupled to the victim, it is not necessary for the output arrival time of $v_o^l(t)$ to be always be greater than that of $v_o^e(t)$.

To find the latest victim-output arrival time, we must allow any feasible victim-input transition occurring within its timing window. Furthermore, for each victim transition, we need to find the alignment of the aggressors such that delay noise is maximized. However, sweeping the victim transition and computing the alignment of aggressors at each point is not feasible, in particular for nonlinear-driver models which require time-consuming nonlinear simulations. As a result, several heuristic solutions [13, 75] have been proposed to avoid an exhaustive enumeration of the victim timing window. In [13], the authors proposed to enumerate the victim alignment at the end

points of victim and the aggressor timing windows. For example, in Figure 2.1, we would require the alignment of the victim transition at four different arrival times t_1, t_2, t_3 and t_v. For each victim alignment, we find the worst-case alignment of all the aggressors and compute the worst-case delay noise. Finally, the aggressor-victim alignment which results in the maximum victim-output arrival time is chosen. The complexity of the above approach is $O(n^2)$, where n refers to number of aggressors coupled to the victim. Since the worst-case alignment of the aggressors has to be recomputed for every victim transition, the above approach can be computationally expensive. Also, such heuristic techniques cannot guarantee the optimality of the results, since we have not considered all feasible victim-input transitions in the timing window (only four in the above example).

In this chapter, we show that the magnitude of slope of the curve in Region B (of Figure 2.2) can never be greater than one. It means that if the victim-input transition is delayed and the delay noise decreases due to misalignment, then this decrease is not sufficient to compensate for the fact that this victim-output transition starts later. Hence, we obtain the useful result that the worst-case victim alignment can occur only when the victim-input transition is aligned at its latest arrival time. Consequently, even with nonlinear aggressor-victim drivers, any search of victim alignment within its timing window is not necessary, and we can safely align the victim-input transition at the latest point in its timing window. Since, we need to compute the worst-case aggressor alignment for only a single victim transition, we obtain a significant speed-up in noise analysis.

While this result has been empirically observed in industrial tools, to the best of our knowledge, this is the first work which proves that the above result holds for both linear and nonlinear driver models. The proof is based on simple properties of

standard nonlinear CMOS drivers as discussed in Section 2.3.1. For simplicity in our analysis, we assume monotonic-input transitions and this assumption is discussed in more detail in Section 2.3.4.

The rest of the chapter is organized as follows: Section 2.2 proves the latest victim-alignment result assuming linear-driver models. Section 2.3 forms the core of this chapter, where we prove this result for the more general case of nonlinear CMOS drivers. It will be shown that the latest victim-input alignment that maximizes the victim-output arrival time also implicitly maximizes the victim *receiver*-output arrival time. In Section 2.4, we extend the proof for the case when the victim is coupled to multiple aggressors. In Section 2.5, we show experimental results that confirm the accuracy and efficacy of the proposed approach, and we summarize the chapter in Section 2.6.

2.2 Victim Alignment for Linear-Driver Models

It is well-known that nonlinear-driver models [47, 70, 24] provide better accuracy in timing analysis than linear-driver models. Nevertheless, linear-driver models are still being used in existing industrial tools [49] for fast analysis in the early stages of design. For linear-driver models, superposition principle can be used to break the cyclic dependency between the aggressor-victim responses and simplify the task of finding the worst-case aggressor-victim alignment. In this section we prove that for linear drivers, the latest victim-output arrival time occurs only when the victim-input transition is aligned at the latest arrival time. We will later review the victim alignment for nonlinear-driver models in Section 2.3.

Theorem 2.1. *Given linear aggressor-victim driver models and monotonic victim-input transitions, the victim-output transition obtained by aligning the victim-input*

transition at the latest victim-input arrival time bounds all possible victim-output transitions.

Proof. Consider the aggressor-victim configuration as shown in Figure 2.3 with linear aggressor-victim drivers. Let $v_i^l(t)$ be the late victim-input transition that is aligned at the latest arrival time in its timing window and $v_i^e(t)$ be any earlier input transition occurring within its timing window. The corresponding victim-output transitions are denoted by $v_o^l(t)$ and $v_o^e(t)$, respectively. Without loss of generality, we assume inverting aggressor-victim drivers and a falling (rising) victim (aggressor) input transition. Our goal is to show that the latest rising victim-output transition $v_o^l(t)$ bounds any earlier victim-output transition,

$$v_o^e(t) \;\geq\; v_o^l(t) \quad \forall t. \tag{2.1}$$

We need to show that Equation 2.1 holds for all feasible aggressor transitions. Therefore, we do not impose any restrictions on the aggressor-input transition and allow it to occur anywhere within its timing window. Applying the principle of superposition, which holds for linear-driver models, the noisy victim-output transition $v_o^l(t)$ can be written as

$$v_o^l(t) \;=\; \bar{v}_o^l(t) + v_n(t), \tag{2.2}$$

where $\bar{v}_o^l(t)$ is the noiseless victim-output transition obtained with a quiet aggressor and $v_n(t)$ is the coupling-noise pulse on the victim. Since the aggressor-input transition is the same for both early-late victim transitions and the victim driver-output resistance is constant, the coupling-noise pulse on the victim $v_n(t)$ is the same in

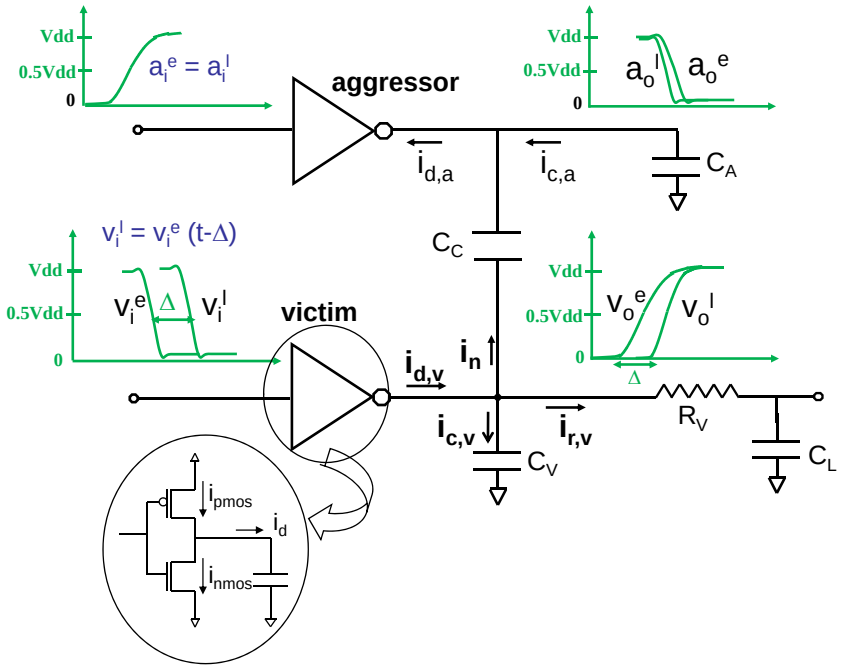


Figure 2.3: Capacitively coupled aggressor-victim nets.

both early and late cases. Therefore, the early victim-output transition $v_o^e(t)$ is given by

$$v_o^e(t) = \bar{v}_o^e(t) + v_n(t). \tag{2.3}$$

From causality arguments, if the separation in time between the victim-input transitions $v_i^e(t)$ and $v_i^l(t)$ is $\triangle$, then the noiseless output waveforms $\bar{v}_o^e(t)$ and $\bar{v}_o^l(t)$ would also be separated by $\triangle$,

$$\begin{aligned} v_i^l(t) &= v_i^e(t - \triangle) \\ \Rightarrow \bar{v}_o^l(t) &= \bar{v}_o^e(t - \triangle). \end{aligned} \tag{2.4}$$

Since the inputs are falling monotonically, the noiseless output transitions must therefore be rising monotonically. As a result, the late noiseless output transition

always bounds the early noiseless output transition

$$\bar{v}_o^e(t) \;\geq\; \bar{v}_o^l(t) \quad \forall t. \tag{2.5}$$

Assuming linear-driver models, the noise $v_n(t)$ remains the same for both early-late victim transitions. Therefore, inserting Equations 2.2-2.3 into the above equation, it follows that the noisy victim-output transition $v_o^l(t)$ bounds $v_o^e(t)$, i.e.,

$$v_o^e(t) \;\geq\; v_o^l(t) \quad \forall t. \tag{2.6}$$

Since the late victim-output transition $v_o^l(t)$ is always less than the early victim-output transition $v_o^e(t)$, it will always cross the 50% (or any other) supply voltage point later than $v_o^e(t)$. Therefore, the latest victim-output arrival time always occurs when the victim-input transition is aligned at the latest-possible input arrival time.

□

2.3 Victim Alignment for Nonlinear-Driver Models

In order to model the nonlinearity of CMOS drivers in noise analysis, nonlinear-driver models such as current-source models [47, 70, 24] provide better accuracy than linear-driver models. In this section, we show that the victim-alignment result derived in the previous section also holds for nonlinear drivers. We begin this section by describing the characteristics of the output current sourced by CMOS drivers.

2.3.1 Properties of Nonlinear Drivers

In order to derive the latest victim-alignment result, we consider a nonlinear inverting CMOS driver where $i_d(t) = f\left(v_i(t), v_o(t)\right)$ is the *steady-state* current flowing

out of the driver, $v_i(t)$ and $v_o(t)$ are the input and output voltages of the driver, respectively. For a rising-output transition (see Figure 2.3), $i_d(t)$ is obtained as the difference between the drive-currents sourced by the pull-up and the pull-down network [56],

$$i_d(t) \;=\; i_{ds}^{pull-up}(t) - i_{ds}^{pull-down}(t). \tag{2.7}$$

From transistor characteristics, we know that given a constant drain-source voltage V_{ds}, the steady-state drain current I_{ds} is a monotonic increasing function of its gate-source voltage V_{gs}. We also know that the V_{gs} of the pull-up and the pull-down network of a driver depends only on the input voltage $v_i(t)$ of the gate,

$$\begin{aligned} V_{gs}^{pull-down} &\propto v_i(t), \\ V_{gs}^{pull-up} &\propto V_{dd} - v_i(t). \end{aligned} \tag{2.8}$$

Therefore, a decrease in the input voltage of the gate will affect the V_{gs} of the pull-up and pull-down network. From basic transistor current-voltage characteristics it follows that, for a *constant* output voltage, a *decrease* in the gate input voltage results in an increase (decrease) in the pull-up (pull-down) current. Therefore, given a constant output-voltage, the steady-state output drive current $i_d(t)$ given by the difference between the pull-up and the pull-down current (see Equation 2.7) also *increases*. In a DC analysis, the above result also holds for more complex gates or gates with skewed transistor stacks.

A similar analysis leads to the observation that, given a constant gate input voltage, a decrease in output voltage leads to an increase (decrease) in the pull-up (down) current, resulting in an increase (decrease) in the output current. Since the

above property relies only on the monotone behavior of the DC current I_{ds} with V_{ds} for MOSFET transistors, it also holds for complex gates with transistor stacks and internal nodes. We sum up the above observations in the following property which relates the driver-output current to driver input-output voltages:

Property 2.2. *Given a specific input and output voltage, the magnitude of the current flowing out of an inverting CMOS driver increases with a decrease in its input or output voltage.*

Strictly speaking, the above properties may not hold true during the entire transition due to the effect of Miller capacitance between the driver input-output nodes, especially when the driver input switches very rapidly. However, one can also note that the amount of Miller current strongly depends on the ratio of the Miller capacitance to the output-load capacitance. Also, delay noise is significant for those victim nets that are coupled to several aggressors and have substantial output loading. For such victim nets, the Miller current is typically negligible compared to driver current and *Property* 2.2 implicitly holds. Furthermore, the Miller current is significant only for very fast input transitions and affects the initial part of the output transition, which is of lesser interest in noise analysis. We will use the above property of the driver-output current to prove the latest victim-input alignment result.

2.3.2 Worst-Case Victim Alignment

In this subsection, we prove by contradiction that when the victim net is coupled to a single aggressor, the latest victim-output transition bounds all possible victim-output transitions. In Section 2.4, we will extend the proof to the more general case of multiple aggressors.

Theorem 2.3. *Given nonlinear victim-aggressor drivers and monotonic victim-input*

transitions, the victim-output transition obtained by aligning the victim-input transition at its latest input arrival time bounds any other feasible victim-output transition.

Proof. Consider the aggressor-victim configuration shown in Figure 2.3, where C_a denotes the output capacitance of the aggressor driver, C_c denotes the coupling capacitance, and the victim driver has a reduced RC pi-model interconnect load. The victim-input transition $v_i^l(t)$ is aligned at the latest time point in its timing window and $v_i^e(t)$ is an arbitrary earlier victim-input transition. The corresponding victim-output transitions are denoted by $v_o^e(t)$ and $v_o^l(t)$, respectively. Our goal is to show that the latest victim-output transition $v_o^l(t)$ bounds any early victim-output transition $v_o^e(t)$ as expressed in Equation 2.1. Without loss of generality, we assume a rising victim and a falling aggressor output transition. Since it is necessary to show that Equation 2.1 holds for all feasible aggressor transitions, we arbitrarily select an aggressor-input transition that can occur anywhere within its timing window. Note that, due to coupling noise, the aggressor output transitions $a_o^e(t)$ and $a_o^l(t)$, corresponding to the early-late victim transitions, may be different even though the input transition is the same $\big(i.e.,\ a_i^e(t) = a_i^l(t)\big)$. We first present an outline of the proof:

1. **Victim-Response Analysis**: Suppose that a later victim-output transition crosses an early victim-output transition. Then, at the crossover point, we obtain a necessary relationship between the corresponding noise currents by analyzing the rate of change of the victim-output response.

2. **Aggressor-Response Analysis**: Next, using this relationship between noise currents and the fact that aggressor-input transition is same in both cases, we compare the relative magnitudes of the aggressor-driver currents, and derive a necessary relationship between the aggressor-output responses.

3. **Charge Conservation**: We then analyze the charge accumulated across the coupling capacitance in both early-late victim transitions, and show that the necessary relationship between aggressor-output responses cannot be satisfied.

We prove by contradiction that the later victim-output transition must always bound any earlier victim-output transition, i.e., $v_o^e(t) \geq v_o^l(t)\ \forall t$.

Victim-Response Analysis

We begin the proof by analyzing the response at the output of the victim driver. Suppose, the converse is true and there exists a time when both victim-output waveforms cross each other for the *first* time at time τ_v (as shown in Figure 2.4),

$$v_o^e(t) = v_o^l(t) \mid_{t=\tau_v} . \tag{2.9}$$

From definition, the late victim-output transition $v_o^l(t)$ starts rising after the early transition $v_o^e(t)$. Therefore, if $v_o^l(t)$ manages to cross $v_o^e(t)$ at time τ_v, then it means that $v_o^l(t)$ must be rising at a faster rate that $v_o^e(t)$ at the time instant τ_v,

$$\frac{\partial v_o^l(t)}{\partial t} > \frac{\partial v_o^e(t)}{\partial t} \mid_{t=\tau_v} . \tag{2.10}$$

We know that the charging current flowing into the victim load C_v is given by $i_{c,v}(t) = C_v \times \frac{\partial v_o(t)}{\partial t}$. Using Equation 2.10, we obtain the following relationship,

$$i_{c,v}^l(t) > i_{c,v}^e(t) \mid_{t=\tau_v}, \tag{2.11}$$

where $i_{c,v}^e(t)$ and $i_{c,v}^l(t)$ are the currents flowing into the victim load C_v corresponding to the early-late victim transitions $v_o^e(t)$ and $v_o^l(t)$, respectively (see Figure 2.3). At the crossover time τ_v, the output voltages of the victim driver must be equal. From the assumption of monotonic falling victim-input transitions, we obtain the inequality $v_i^e(t) \leq v_i^l(t) \mid_{t=\tau_v}$. It follows from *Property* 2.2 that the victim driver sources a larger

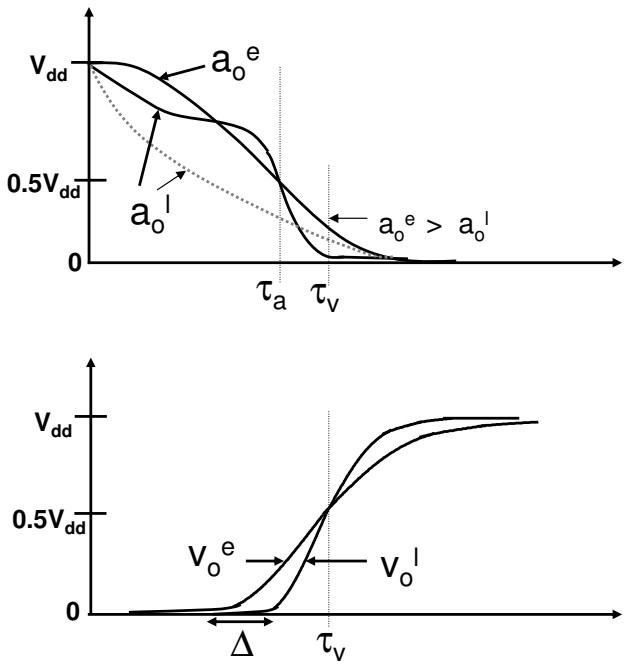


Figure 2.4: Late victim-output transition crossing an early victim-output transition.

output current in the case of an early victim transition as compared to the late victim transition,

$$i^e_{d,v}(t) \; > \; i^l_{d,v}(t) \;\; |_{t=\tau_v} \; . \tag{2.12}$$

It can be proved that the following relationship holds (see *Appendix A*) between the currents flowing into the victim load C_L

$$i^l_{r,v}(t) \; > \; i^e_{r,v}(t) \;\; |_{t=\tau_v} \; . \tag{2.13}$$

We can combine the inequalities in Equations 2.11, 2.12 and 2.13 to obtain the following inequality

$$i^e_{d,v}(t) - i^e_{c,v}(t) - i^e_{r,v}(t) \; > \; i^l_{d,v}(t) - i^l_{c,v}(t) - i^l_{c,v}(t) \;\; |_{t=\tau_v} \; . \tag{2.14}$$

Applying Kirchhoff's Current Law (K.C.L.) at the victim-output node, and rewriting Equation 2.14 in terms of the corresponding noise currents flowing through the coupling capacitance C_c, we obtain

$$i_n^e(t) \; > \; i_n^l(t) \; |_{t=\tau_v} \; . \tag{2.15}$$

Aggressor-Response Analysis

Using the information about the victim-output transitions, we obtain a relationship between the early-late aggressor output waveforms. The noise current flowing through the coupling capacitance C_c can be expressed in terms of the rate of change of the voltage difference across its terminals. After plugging the expressions for noise current into Equation 2.15, we obtain

$$C_c \times \frac{\partial\{v_o^e(t) - a_o^e(t)\}}{\partial t} \; > \; C_c \times \frac{\partial\{v_o^l(t) - a_o^l(t)\}}{\partial t} \; |_{t=\tau_v}, \tag{2.16}$$

where $a_o^e(t)$ and $a_o^l(t)$ are the corresponding early-late aggressor-output waveforms (see Figure 2.3). Rearranging both sides of Equation 2.16, we obtain

$$\frac{\partial\{a_o^l(t) - a_o^e(t)\}}{\partial t} \; > \; \frac{\partial\{v_o^l(t) - v_o^e(t)\}}{\partial t} \; |_{t=\tau_v} \; . \tag{2.17}$$

From Equation 2.10, we know that at crossover time τ_v, the rate of change of the late victim transition $v_o^l(t)$ is greater than that of $v_o^e(t)$. Therefore, the term on the right hand side of Equation 2.17 must be greater than zero, and we get the following inequality between the early-late aggressor-output waveforms

$$\frac{\partial a_o^l(t)}{\partial t} > \frac{\partial a_o^e(t)}{\partial t} \mid_{t=\tau_v} . \tag{2.18}$$

We provide an intuition for the analytical results obtained. Suppose, $v_o^l(t)$ rises at a relatively faster rate and crosses $v_o^e(t)$ at time τ_v, which implies two things about the relative magnitudes of the victim currents at the time instant τ_v. For the late victim-output transition $v_o^l(t)$, (1) the current sourced by the victim driver is less, and (2) the charging current flowing into the interconnect load is more, compared to the early victim transition $v_o^e(t)$. Since the current sourced by the victim driver equals the sum of the noise current and the charging load current, at time τ_v, the noise current must be less for the later victim transition. Also, the noise current depends on the rate of change of the voltage difference across the coupling capacitance. At time τ_v, the rate of change of the later victim-output transition is more. Therefore, the relationship among the noise currents demands that the rate of change of $a_o^l(t)$ be less than that of $a_o^e(t)$. Note that the aggressor has a falling output transition. Therefore, both derivative terms in Equation 2.18 are negative in magnitude, and early aggressor $a_o^e(t)$ falls more rapidly than $a_o^l(t)$. The discharging current of the aggressor interconnect load is given by

$$i_{c,a}(t) = -C_a \times \frac{\partial a_o(t)}{\partial t}. \tag{2.19}$$

The negative sign in the above expression is due to the convention followed that the load current is flowing out of the load capacitance C_a (see Figure 2.3). Using Equations 2.19-2.18, we obtain the following inequality between the early-late

aggressor interconnect load currents

$$i^{e}_{c,a}(t) \;>\; i^{l}_{c,a}(t) \;\;|_{t=\tau_v} \;. \tag{2.20}$$

Adding Equations 2.20-2.15 and applying K.C.L. at the aggressor-output node, we obtain the following inequality among the aggressor-driver currents,

$$i^{e}_{d,a}(t) \;>\; i^{l}_{d,a}(t) \;\;|_{t=\tau_v} \;. \tag{2.21}$$

Since the aggressor-input transitions are the same in both cases, $\big(i.e.,\; a^{e}_{i}(t) = a^{l}_{i}(t)\big)$, the gate voltages of the aggressor driver are equal. Now, if the aggressor-driver currents differ according to Equation 2.21, then from *Property* 2.2 it follows that the output voltages of the aggressor must satisfy the following relationship,

$$a^{e}_{o}(t) \;>\; a^{l}_{o}(t) \;\;|_{t=\tau_v} \;. \tag{2.22}$$

To summarize, we obtain two necessary conditions on the aggressor output waveforms at time τ_v, i.e., $a^{l}_{o}(t)$ must be lower and must transition at a slower rate than $a^{e}_{o}(t)$ (from Equations 2.22-2.18).

Charge-Conservation Analysis

We will analyze the relationships between the driver currents that is sunk by the aggressor driver. Hence, we need to define a time interval during which we compute the amount of charge sunk by aggressor driver. It follows from Equation 2.22 that there must be a time $\tau_a : 0 \leq \tau_a < \tau_v$ where the early-late aggressor output transitions intersect each other (see Figure 2.4)

$$a^{e}_{o}(t) \;=\; a^{l}_{o}(t) \;\;|_{t=\tau_a} \;. \tag{2.23}$$

If the late aggressor output transition $a_o^l(t)$ is always less than $a_o^e(t)$, then we obtain $\tau_a = 0$ (as shown by the dotted waveform). If $a_o^l(t)$ and $a_o^e(t)$ have multiple crossovers before time τ_v, then we choose τ_a to be the time at which the latest crossover occurs between the aggressor output transitions. The boundary conditions on the aggressor-victim output transitions in the interval $[\tau_a, \tau_v]$ are as follows (as shown in Figure 2.4)

$$\begin{array}{lllll} a_o^e(\tau_a) & = & a_o^l(\tau_a) & , & a_o^e(\tau_v) > a_o^l(\tau_v) \\ v_o^e(\tau_a) & > & v_o^l(\tau_a) & , & v_o^e(\tau_v) = v_o^l(\tau_v). \end{array} \tag{2.24}$$

Since we chose τ_a to be the latest crossover time of the aggressor-output transitions before τ_v, we obtain the following monotonic relationship between the aggressor output transitions,

$$a_o^e(t) > a_o^l(t) \quad \forall t \in (\tau_a, \tau_v). \tag{2.25}$$

Recall that the input to the aggressor driver is the same in both cases. From *Property* 2.2 and Equation 2.25, it follows that the current sunk by the aggressor driver in the time interval is always greater in the early case, i.e.,

$$i_{d,a}^e(t) > i_{d,a}^l(t) \quad \forall t \in (\tau_a, \tau_v). \tag{2.26}$$

Integrating the above, we obtain the inequality between the total charge sunk by the aggressor driver Q_d^e and Q_d^l, respectively,

$$\begin{array}{lll} \int_{\tau_a}^{\tau_v} i_{d,a}^e(t) \cdot dt & > & \int_{\tau_a}^{\tau_v} i_{d,a}^l(t) \cdot dt \\ \Longrightarrow \quad Q_d^e & > & Q_d^l. \end{array} \tag{2.27}$$

We analyze the relationship between the integral of the noise current $i_n(t)$ and that of the load current $i_{v,a}(t)$ flowing into the aggressor in the time interval (τ_a, τ_v). As both integrals are state functions, they do not depend on the integration path and only depend on the voltage values at the boundaries of the interval (τ_a, τ_v). The integral of load current $Q^e_{c,a}$ for the early victim transition is given by

$$Q^e_{c,a} = \int_{\tau_a}^{\tau_v} i^e_{c,a}(t) \cdot dt = -C_a \times \{a^e_o(\tau_v) - a^e_o(\tau_a)\}, \tag{2.28}$$

and similarly the integral of noise current Q^e_n is given by

$$Q^e_n = \int_{\tau_a}^{\tau_v} i^e_n(t) \cdot dt = C_c \times \{v^e_o(\tau_v) - v^e_o(\tau_a) + a^e_o(\tau_a) - a^e_o(\tau_v)\}. \tag{2.29}$$

Similarly, we can derive the integral of the load current $Q^l_{c,a}$ and the integral of the noise current Q^l_n for the late victim transition. After plugging in the boundary values from Equation 2.24, we obtain the following inequalities,

$$Q^l_{c,a} > Q^e_{c,a} \quad and \quad Q^l_n > Q^e_n. \tag{2.30}$$

Next, we add both inequalities in Equation 2.30 and apply K.C.L on the aggressor-output node to obtain the following inequality

$$Q^l_d > Q^e_d, \tag{2.31}$$

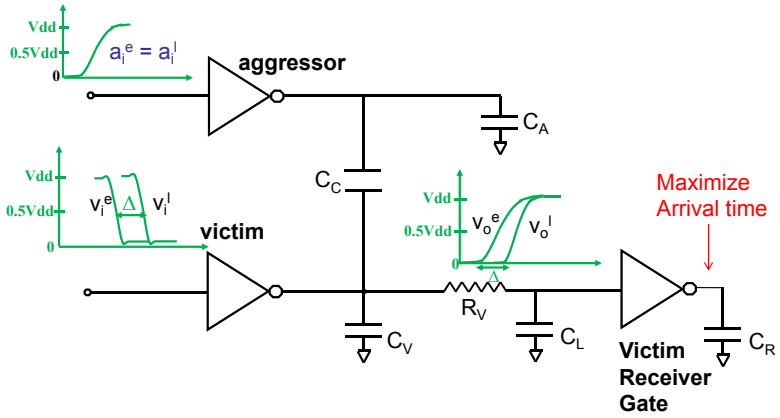


Figure 2.5: Maximize the victim receiver-output arrival time.

which contradicts the necessary condition obtained in Equation 2.27. Therefore, we prove by contradiction that the victim-output transition $v_o^e(t)$ can never cross $v_o^l(t)$, and that the latest victim-output arrival time occurs only when the victim-input transition is aligned at its latest arrival time. □

2.3.3 Worst-Case Victim-Receiver Output Alignment

In this chapter, we focused on finding the victim-input transition which results in the latest victim-output arrival time. However, as reported in [47], the true objective of noise analysis is not to maximize the victim-output arrival time, but to maximize the output arrival time of the victim *receiver* gate (see Figure 2.5). In this subsection we show that the victim alignment at the latest point in its timing window, also maximizes the victim receiver-output arrival time.

Theorem 2.4. *Given nonlinear aggressor-victim driver models and monotonic victim-input transitions, alignment of the victim-input transition at the latest input arrival time results in the latest arrival time at the output of the victim receiver.*

Proof. Given nonlinear aggressor-victim drivers, it was proved that the early $(v_o^e(t))$ and late $\left(v_o^l(t)\right)$ victim-output waveforms do not cross each other. Hence, the latest

victim-output transition always bounds the early victim-output transition. It can be noted that we have exactly the same setup at the input of the victim *receiver* gate as we had for the victim *driver* gate and $v_o^l(t)$ bounds $v_o^e(t)$ at the input of the victim receiver. Therefore, using *Property* 2.2, we can compare the relative magnitudes of the driver-output current sourced in both cases. Repeating the exact same analysis, which was performed on the victim driver, for the victim-receiver gate we can arrive at the desired result which claims that late victim receiver-output transition will bound any earlier victim-output transition. □

2.3.4 Monotonic-Input Transitions

A necessary condition for the latest victim-alignment result was that the latest victim-input transition always bounds any earlier input transition. Using *Property* 2.2, we then compare the relative magnitudes of the driver-output current sourced in both cases. The assumption of monotonic victim-input transitions is a more restrictive condition which ensures that the latest victim-input transition always bounds an earlier input transition. With nonmonotonic input waveforms, the latest victim-input transition may no longer bound any earlier victim-input transition. It is not clear whether it can actually result in a later victim-output arrival time for an earlier victim-input transition. However, in practice, it should be noted that nonmonotonic transitions filter out rapidly due to the low-pass filtering effects of CMOS drivers. Therefore, one can apply the latest victim-alignment result to only those cases where the input transitions are monotonic and still obtain significant speed-up in noise analysis.

2.4 Victim Alignment for Multiple Aggressors

In the previous section, we show that the latest victim-alignment result holds for the case when the victim was coupled to a single aggressor. However, the victim net is usually coupled to more than one aggressors. Therefore, it is natural to ask whether the latest victim-alignment result also holds for the case when the victim is coupled to multiple aggressors. In this section, we state and prove the following theorem,

Theorem 2.5. *Given nonlinear aggressor-victim drivers and multiple aggressors coupled to the victim net, the victim-output transition obtained when its input-transition occurs at the latest input arrival time bounds any other victim-output transition.*

Proof. Suppose, there are K aggressors coupled to the victim net, and the noise current flowing into the k^{th} aggressor be given by $i_{n,k}(t)$ (as shown in Figure 2.6). We perform the *victim-response analysis* by first assuming that both early-late victim-output waveforms cross each other at time τ_v. Now, the total noise current flowing out of the victim interconnect will be the cumulative sum of the individual noise

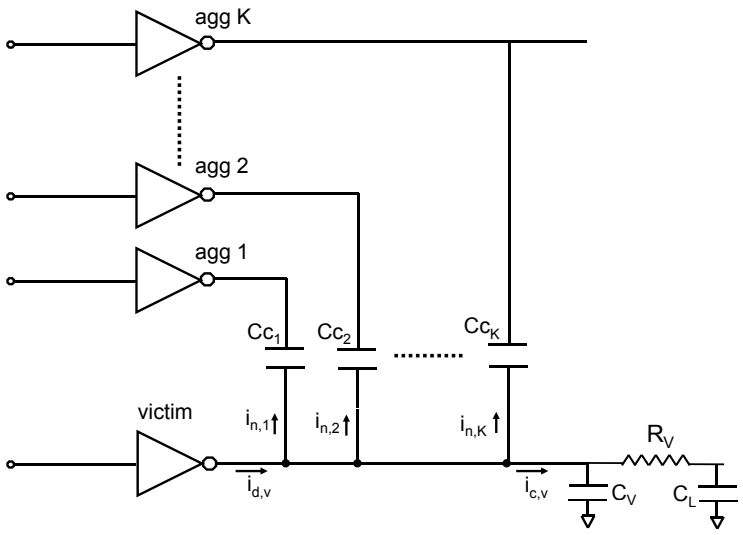


Figure 2.6: Victim net coupled to multiple aggressors.

currents flowing through each coupling capacitance. Therefore, the current flowing into the victim interconnect ($i_{c,v}(t)$) can be obtained by the subtracting all the noise currents from the victim-driver current ($i_{d,v}(t)$),

$$i_{c,v}(t) \quad = \quad i_{d,v}(t) - \sum_{k=1}^{k=K} i_{n,k}(t). \tag{2.32}$$

We substitute the expression of $i_{c,v}(t)$ derived above into Equations 2.11-2.12 to obtain the following inequality between the cumulative sum of the early-late noise currents at crossover time τ_v,

$$\sum_{k=1}^{k=K} i^e_{n,k}(t) \quad > \quad \sum_{k=1}^{k=K} i^l_{n,k}(t) \;\; |_{t=\tau_v} \; . \tag{2.33}$$

Note that the above inequality between the noise currents is a necessary condition for a crossover to occur between the early-late victim-output transitions at time τ_v. For the inequality in Equation 2.33 to hold, there must be at *least* one aggressor, say a_m, whose noise currents satisfies the following relationship

$$i^e_{n,m}(t) \quad > \quad i^l_{n,m}(t) \;\; |_{t=\tau_v} \; . \tag{2.34}$$

The analysis that follows is exactly the same as that for the single aggressor case described in the previous section. Performing the *aggressor-response analysis* on a_m, we obtain the following relationships between the early-late aggressor output waveforms $a^e_{o,m}(t)$ and $a^l_{o,m}(t)$ at the crossover time τ_v

$$a^e_{o,m}(t) \quad > \quad a^l_{o,m}(t) \;\; |_{t=\tau_v} \; . \tag{2.35}$$

Proceeding in a similar fashion, we identify the latest crossover time $\tau_{a,m}$ between the aggressor-output waveforms ($a^e_{o,m}$ and $a^l_{o,m}$) occurring before the time τ_v. Finally, we perform *charge conservation analysis* on the aggressor a_m. We compare the magnitude of the total charge sunk through the aggressor driver within the time interval ($\tau_{a,m}$, τ_v) in the early-late cases and obtain the desired contradiction. □

It was shown that even when the victim is coupled to multiple aggressors, the early-late victim-output transitions can never cross each other. Therefore, the latest victim-output arrival time occurs only when its input is aligned at the latest point in its timing window. Note that while computing the worst-case victim-output arrival time for a given aggressor-victim configuration, we account for only the mutual interaction between the victim and the aggressor transitions. In the noise analysis, the mutual interaction between the aggressor transitions are captured during the global iterations of the delay-noise computation algorithm.

2.5 Experimental Results

In this section, we look at experimental results which confirm the accuracy and the efficacy of our proposed approach by reporting the runtime improvement over existing approaches.

2.5.1 HSPICE Simulations

To further illustrate the ideas put forward in this chapter, in Figure 2.7, we show the output responses generated in HSPICE for coupled aggressor-victim nets. We fix the aggressor-input arrival time and sweep the input skew by shifting the victim-input transition to the right. Note that as we delay the victim-input transition,

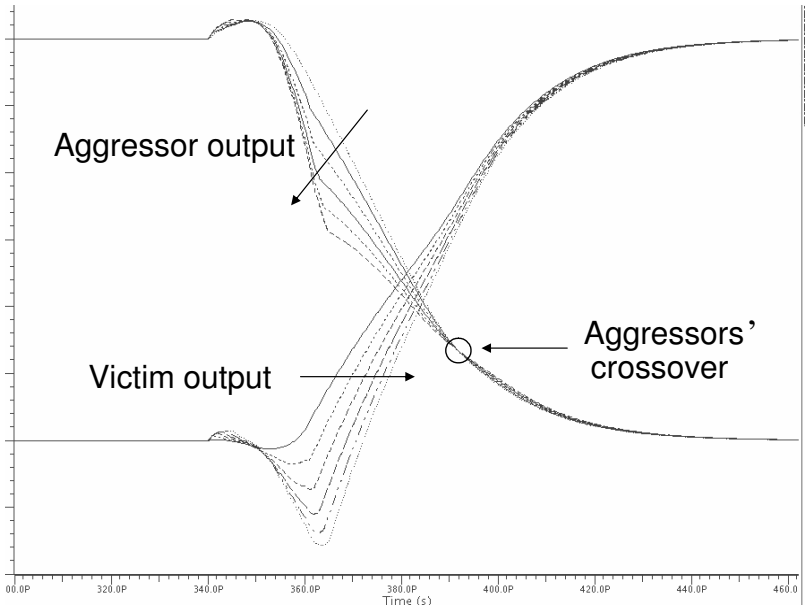


Figure 2.7: HSPICE simulation plots showing the aggressor-victim output transitions.

the aggressor-output initially starts to transition faster because the effect of Miller-coupling, which occurs due to the switching of the victim, is correspondingly delayed. Also, due to the noise coupled from the aggressor, the delayed victim-output transition starts from a voltage below zero and increases the drain-source voltage of the pull-up network of the victim driver, and the victim-output waveform starts rising rapidly. However, as the victim-output transition starts approaching an earlier transition, the corresponding aggressor output transitions cross each other (see Figure 2.7) and violates the necessary condition for a crossover to occur between the early-late victim-output transitions (Equation 2.22). Hence, the victim-output transitions never cross each other.

2.5.2 Delay-Noise Analysis

A prototype noise-analysis tool was implemented in C++ programming language, and uses linear-driver models to perform the noise analysis. This tool uses the

Table 2.1: Results for the proposed latest victim-alignment approach.

ckt	#nets	#agg	ckt delay(*in ns*)	#agg pruned	%agg pruned	run time(*in s*)	speed-up
i1	46	232	0.546	103	44.39	0.01	2.74X
i2	221	706	0.743	324	45.89	0.02	2.46X
i3	126	551	0.529	281	50.99	0.02	3.12X
i4	230	1181	0.801	610	51.65	0.02	3.56X
i5	138	1835	1.212	794	43.27	0.04	4.88X
i6	668	7298	1.045	3066	42.01	0.14	5.15X
i7	870	9605	1.124	4925	51.27	0.15	6.19X
i8	1528	10235	1.636	5436	53.11	0.21	5.32X
i9	955	14140	1.841	6789	48.02	0.33	6.91X
i10	3155	18318	3.089	8744	47.73	0.45	3.21X

proposed approach of aligning the victim transition at the latest point in its timing window and was tested on the LGSynth91 benchmark circuits [1]. We used up to three metal layers while doing the place-and-route of the circuits with the target utilization of the floorplan set at 80%. The Cadence Silicon Ensemble tool was used to perform the place-and-route of the benchmark circuits. The Mentor Graphics Calibre tool was used to extract the parasitic coupling and ground capacitances for the nets in the design.

A summary of the experimental results obtained for the LGSynth91 benchmark circuits is listed in Table 2.1. The circuit details are given in the first four columns of Table 2.1, and the results of the proposed approach are given in the final four columns. To obtain the worst-case victim-output arrival time, we align the victim-input transition at the latest point in its timing window. The worst-case aggressor alignment is computed such that it maximizes the 50% crossover time of the victim transition [37]. The fact that we no longer need to search for the victim alignment within the victim timing window simplifies the overall alignment algorithm and results in a speed up of the noise-analysis engine. We compare the latest victim-alignment approach with that proposed in [13], where the victim alignment is enumerated at

the end points of the aggressor-victim timing windows. For example, in Figure 2.1, this approach requires the alignment of the victim transition at four different arrival times t_1, t_2, t_3 and t_v. The worst-case alignment of aggressors was found for each victim alignment, and finally the aggressor-victim alignment that results in the maximum output arrival time was reported. We observed that the delay noise obtained by both approaches were identical, which confirms the fact that the victim alignment at the latest point in its timing window is optimal, and an enumeration of the victim alignment within its timing window is not necessary to maximize the victim-output arrival time. Since our proposed approach requires only a single victim alignment, an average speed up of $4.3X$ was achieved on benchmark circuits.

With the victim-input alignment fixed at the end points of its timing window, the number of aggressors that can align with the victim transition are reduced substantially. For example, in Figure 2.1, only aggressor a_3 can inject noise on the victim transition. All aggressors that can no longer inject noise on the fixed victim transition (e.g., aggressors a_{1-2} in Figure 2.1) can be safely ignored in noise analysis due to temporal filtering. For the benchmark circuits, approximately half of the total number of aggressors in the circuit can be eliminated from noise analysis due to temporal filtering as shown in Column six of Table 2.1, which results in a further speed up of the overall noise-analysis engine. We can also note that the maximum speed-up is achieved for circuit *i9* which has the largest number of aggressors per victim net, i.e., approximately 14 aggressors per victim. Hence, a larger speed-up can be achieved for larger industrial circuits with several aggressors nets per victim.

2.6 Summary

In this chapter, we proved that the latest victim-output arrival time occurs only when its input transition is aligned at the latest point in its timing window. While the proof was fairly straightforward for linear-driver models, it was certainly nontrivial for nonlinear CMOS drivers. This result would obviate the need for enumerating the victim-input timing window in noise analysis. Consequently, the aggressor-victim alignment problem is simplified and its complexity is reduced significantly. Although this result has been observed empirically in the industry, this is the first work which analytically shows that the result holds for both linear and nonlinear drivers. Using this result, we show that significant speed-up can be achieved on benchmark circuits over existing heuristic solutions without incurring any loss of accuracy.

CHAPTER III

Worst-Case Aggressor-Victim Alignment with Current-Source Drivers

Continuous scaling of device dimensions in the nanometer process technology has led to several key challenges in the timing verification of circuits. As the spacing between adjacent wires continues to shrink, the coupling capacitance dominates the total wire capacitance. Furthermore, as technology advances, we are seeing increasing chip frequencies and decreasing voltage margins. All of the above trends exacerbate crosstalk noise which occurs due to the charge transfer between the coupled interconnects. Therefore, it has become imperative to accurately model the impact of crosstalk noise on circuit delay while performing timing analysis for nanometer VLSI circuits.

Besides crosstalk noise, we have several nonlinear effects that must be modeled to accurately estimate the gate and interconnect delays. These effects include multiple-input switching, resistive and inductive interconnects, power-grid noise, and nonlinear gate capacitances, etc. Traditionally, cell delays have been computed by using look-up tables or k-factor equations, where the output load is usually a lumped capacitance, and the input transition is approximated by a ramp or other characterization-waveform shapes [55, 26, 11]. In this framework, an effective-capacitance (*Ceff*) based technique was developed in [25] to model the resistive-shielding effect ob-

served in distributed interconnects. In [15, 57], the authors obtain a more accurate estimate of the crosstalk-noise pulse by accounting for the nonlinearity of victim-driver resistance. A new gate-delay model was proposed in [21] to account for the increase in gate delay due to the simultaneous switching of gate inputs. However, the above modifications are mostly ad-hoc in nature and may not be very accurate when they are all combined together to perform timing verification of nanometer designs. In contrast, current-source models (CSM) have emerged as a more fundamental approach for performing timing analysis, since they are independent of precharacterized ramp-input waveforms and lumped capacitive output loads. In [24], a CSM was proposed where the output current depends on the DC voltage levels of the input and output pins, and an extra capacitance was added to the output pin to account for the transient effects. In [47], it was shown that CSMs can be effectively used for performing noise analysis.

Traditionally, the objective of noise analysis has been to maximize (or minimize for MIN analysis) the victim-stage delay. Under the linear-superposition assumption, it was shown in [37] that the victim-stage delay is maximized, for a rising victim transition, when the peak of the coupling-noise pulse (Vp) is aligned at the point where the noiseless-victim waveform crosses the $0.5V_{dd} + Vp$ voltage level. However, the true objective of noise analysis is not to maximize the victim-stage delay, but to maximize the combined sum of the victim-stage delay and the victim-receiver stage delay. In other words, the worst-case alignment of aggressors should maximize the output arrival time of the victim receiver. We show with an example that maximizing the former quantity does not guarantee the maximization of the latter. Figure 3.1 illustrates coupled aggressor-victim nets with a lightly loaded victim-receiver gate. The victim-input transition is fixed, and the alignment of the aggressor-input transi-

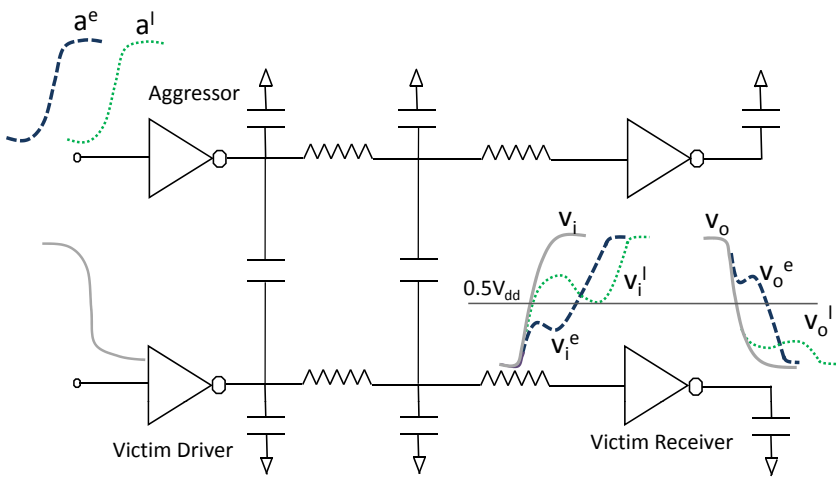


Figure 3.1: Aggressor alignment maximizes the victim-stage delay but not the victim receiver-output arrival time.

tion is varied. The noisy-victim waveforms v_i^e and v_i^l correspond to the two aggressor transitions a_i^e and a_i^l, respectively. It is can be noted that v_i^l has a later 50% crossing time (t_{50} or *arrival time*) and consequently a greater stage delay than v_i^e. However, the victim transition with the maximum stage delay (v_i^l) results in the output transition v_o^l, for which the coupling-noise pulse arrives too late after it has finished switching. In comparison, the earlier victim transition v_i^e results in the victim-output transition v_o^e having the latest arrival time. Hence, maximizing only the victim-stage delay can sometimes be inaccurate, and the worst-case aggressor alignment should be computed such that it maximizes the victim receiver-output arrival time.

It was shown in [15] that the worst-case aggressor alignment is a nonlinear function of the victim slew rate, coupling-noise pulse and victim receiver-output load. The authors propose the use of precharacterized look-up tables to identify the worst-case aggressor alignment which requires additional overhead in cell-library characterization. In [47], techniques of constrained optimization were used to obtain the worst-case aggressor-victim alignment with the objective of maximizing the victim receiver-output arrival time. However, it may require multiple nonlinear simulations

and can be expensive in terms of runtime overhead. In this work, we present a heuristic method to find the worst-case aggressor-victim alignment considering nonlinear CMOS drivers. We propose the *cumulative gate overdrive voltage* ($CGOV$) metric which models the total victim receiver-output current sourced. We know that the rate of the victim receiver-output transition is proportional to the amount of current sourced by the victim-receiver gate. Hence, the alignment with the lowest $CGOV$ will result in the slowest output transition having the latest arrival time. Using the $CGOV$ metric, we propose a heuristic approach to compute the worst-case aggressor alignment that maximizes the victim receiver-output arrival time. Since the victim receiver-output arrival time is estimated without actually simulating the output waveform, the proposed approach proves to be very runtime efficient. HSPICE simulations performed on industrial nets to validate the proposed methodology show an average error of 1.7% in delay noise compared to the worst-case alignment obtained by an exhaustive sweep.

3.1 Problem Description

In this section, we analyze the problem of computing the worst-case delay noise in a static timing analysis (STA) framework, where every net has a timing window representing the period within the clock cycle within which the net can switch. In [34], it was shown that victim alignment at the latest point in its timing window was optimal and always resulted in the latest victim-output arrival time. In this work, we focus on the problem of computing the alignment of aggressors relative to the victim transition such that they satisfy their respective timing window constraints and produce the maximum delay noise at victim-output transition. It must be noted that the problem of computing delay noise and timing windows are mutually depen-

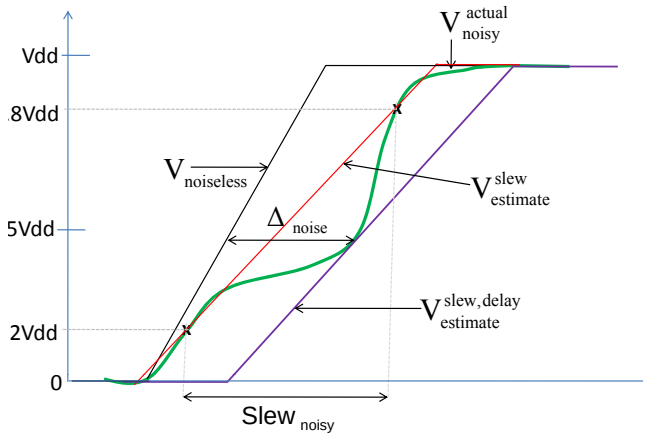


Figure 3.2: Waveforms at the input of the victim-receiver gate.

dent, since the delay noise depends on aggressor timing windows, and the timing window of any net depends on the delay noise. However, it was shown in [61, 12, 77] that this *chicken-and-egg* problem could be solved by updating the delay noise and timing windows iteratively. In this chapter, we consider the problem of computing the worst-case delay noise, given the aggressor timing windows at some iteration of the outer loop.

As seen earlier, the worst-case delay noise should maximize the victim-output arrival time and not just the victim-stage delay. The computation of victim-output transition requires two steps, first the noisy waveform is computed at the input of the victim receiver, second the victim-receiver gate is simulated with the noisy input waveform. In cases where the victim net has a significant amount of coupling, the shape of the noisy-victim waveform (e.g., V_{noisy}^{actual} in Figure 3.2) can be significantly different from a ramp. [1] Such noisy-victim waveforms cannot be used directly as inputs while simulating the victim-receiver gate with traditional look-up table based cell-delay models, which are characterized with ramp input waveforms. Instead, an *equivalent ramp* is often fitted to noisy-victim waveform and used as the input

[1] For simplicity of discussion, we will use a ramp as the representative waveform for all characterization-waveform shapes.

transition for the victim-receiver gate. Several heuristic approaches can be used to obtain an equivalent-ramp signal. One approach matches the slew rate of the noisy-victim transition by fitting a ramp ($V_{estimate}^{slew}$ in Figure 3.2) through the 20-80% VDD voltage trip points of V_{noisy}^{actual}. For more conservative noise analysis, the above ramp can delayed ($V_{estimate}^{slew,delay}$) such that its arrival time matches that of V_{noisy}^{actual}. It can be seen in the figure that $V_{estimate}^{slew,delay}$ underestimates the actual waveform V_{noisy}^{actual} and results in a pessimistic victim-output arrival time. Alternatively, the authors in [39] propose the use of an equivalent ramp that is closest to the noisy-victim waveform in a weighted least-square sense. In [28], the authors obtain a *transition quantity* by integrating the area beneath the noisy-victim waveform and use it as a metric to obtain an equivalent-ramp waveform.

With traditional look-up table based delay models, it is not possible to compute the *exact* victim-output waveform with arbitrary input waveforms. Although, aggressor alignment that maximizes only the victim-stage delay can be optimistic, the use of equivalent ramp such as $V_{estimate}^{slew,delay}$ often guarantees an extra pessimism in delay noise. Overall, the use of traditional look-up table based delay models can lead to erroneous results due to the inherent modeling approximations associated with it, especially when there is a significant amount of coupling noise, and the shape of the victim waveform significantly differs from a ramp signal. In contrast, using current-source models (CSMs), accurate victim-output waveform can be computed even with arbitrary noisy input waveforms (e.g., V_{noisy}^{actual}). We already know that the aggressor alignment that maximizes the victim-stage delay may not necessarily maximize the output arrival time. Therefore, to prevent *optimistic* noise analysis in the CSM framework, we must instead find the alignment that maximizes the victim receiver-output arrival time.

Although, CSMs provide better accuracy over look-up table based delay models, they are computationally more expensive. Also, for victim nets with relatively small amount of coupling noise, noise analysis with traditional models provide sufficient accuracy. Therefore, using CSMs for such cases will incur an additional runtime penalty without significantly improving the accuracy of noise analysis. In order to maximize the accuracy of delay-noise engine without significantly adding a runtime penalty, we propose the following noise-analysis methodology (as shown in Figure 3.3). For the commonly occurring case of very small (e.g., $\leq$ 5% VDD) coupling-noise pulse, we propose to use look-up table based delay models which are very fast and provide reasonable accuracy. Note that if the coupling-noise pulse is very small, then the amount of delay noise would correspondingly be small. Therefore, we do not require a very accurate delay-noise engine in this region, and the aggressor alignment can be computed very quickly by maximizing only the victim-stage delay [12]. However, for victim nets with relatively larger coupling-noise pulses, we propose the use of CSMs to accurately model the noisy-victim waveforms, and the worst-case alignment of aggressors is then computed such that it maximizes the victim-output arrival time. In this work, we present a heuristic approach which accurately computes the worst-

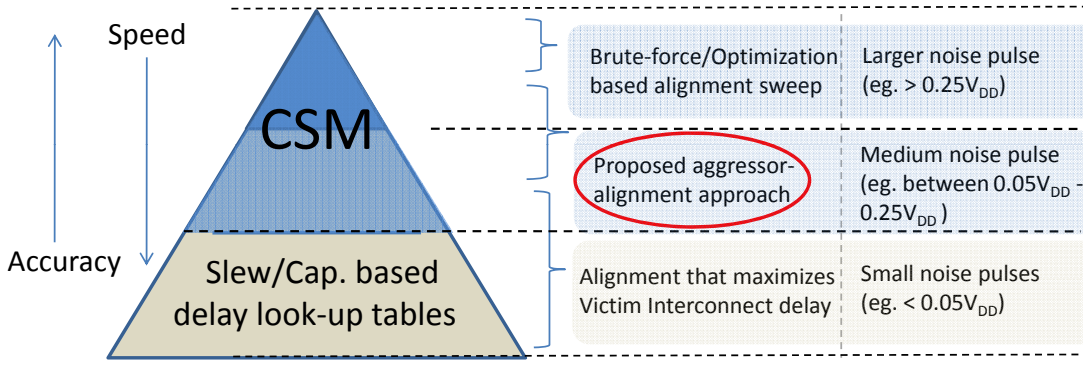


Figure 3.3: Proposed methodology of performing noise analysis.

case aggressor alignment for relatively larger coupling-noise pulses (e.g., [5%, 25%] VDD). We suggest, for even larger coupling-noise pulses (e.g., $\geq$ 25% VDD), the use of more accurate optimization techniques to compute the worst-case alignment. However, with modern place-and-route tools, it is uncommon to have a large number of victim nets with such high amounts of coupling noise. Hence, we need to use the computationally-expensive engine only on a very small fraction of the entire design. Overall, we believe that the proposed methodology produces accurate results with very fast runtime.

The rest of the chapter is organized as follows: In Section 3.2, we first explain the metric that can be used as a proxy for the total victim receiver-output current. In Section 3.3, we show how this metric can be used to compute the worst-case aggressor alignment for both MIN and MAX analysis. In Section 3.4, we present experimental results and compare the delay noise obtained by using our proposed methodology with that obtained by doing worst-case input alignment. We finally summarize this chapter in Section 3.5.

3.2 Cumulative Gate Overdrive Voltage

In this section, we propose a metric to model the total victim receiver-output current. The key observation is that the rate of victim-output transition directly depends on the amount of current sourced by the receiver gate. Hence, we can use the total victim receiver-output current as a proxy for the victim-output transition. Therefore, we can solve for the *latest* occurring victim-output transition without even simulating the victim-output transition. We first summarize the relationship between input voltage and the output current sourced by a CMOS inverter for a rising output transition

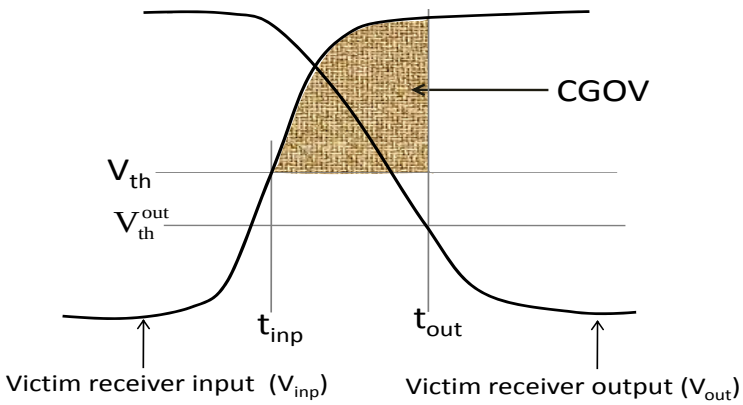


Figure 3.4: Cumulative Gate Overdrive Voltage ($CGOV$).

- The output current sourced by the driver is negligible when the gate input voltage (V_{inp}) is less than the threshold voltage (V_{th})
- The output current sourced by the pull-up network of the driver is proportional to the gate *overdrive* voltage $(V_{inp} - V_{th})^{\alpha}$, for some $\alpha \in (1, 2)$ [44]

In this work, the total output current sourced by the CMOS driver is modeled by the *cumulative gate overdrive voltage* ($CGOV$) which is defined as the area between the input waveform and the threshold voltage of the gate [28] as shown in Figure 3.4.

$$CGOV = \int_{t_{inp}}^{t_{out}} (V_{inp} - V_{th})^{\alpha}\, dt \tag{3.1}$$

Note that t_{inp} is the time when V_{inp} crosses the threshold voltage of the gate V_{th}, which is a function of the transistors in the pull-up (pull-down) stack for a rising (falling) output transition. Similarly t_{out} is the time when the output waveform V_{out} crosses the target voltage level (V_{th}^{out}) of the output loading gate. Therefore, $CGOV$ actually models the total output current that is required to switch the output waveform to the level of the target voltage level V_{th}^{out}.

One can note that $CGOV$ for a certain gate is only a function of the output load, since it tracks the amount of output current that must be sourced for the output transition to switch up to a certain voltage level. In order to accurately compute $CGOV$, we need to model the dependence of output current on the gate output voltage, and must also account for the effects of parasitic Miller-capacitance between input and output nodes of the gate. However, we show that even when the above mentioned second order effects are not modeled in Equation 3.1, the $CGOV$ tracks the arrival time of noisy victim-output transition very closely.

Consider the aggressor-victim coupled network shown in Figure 3.1. First, we fix the victim-receiver gate and sweep all the other circuit parameters, such as aggressor-victim slew rates, driver sizes, and interconnect capacitances by randomly assigning their values. Then, we perform HSPICE simulations on each circuit to obtain the corresponding noiseless (v_i) and the noisy (v_i^l) victim transitions. The histogram consisting of about 1500 data points was obtained for the victim slew rates and coupling-noise peaks as shown in Figure 3.5. Finally, for every circuit, the respective $CGOV$ values were calculated by using Equation 3.1 and integrating up to the arrival times of the corresponding victim-output transitions, v_i and v_o, respectively. The gate threshold voltages were assumed to be 0.5VDD. Shown in Figure 3.5, is a scatter plot of $CGOV$ values obtained for the circuits. On the X (Y) axis, we plot the percentage error of $CGOV$ values with respect to the mean for the corresponding noiseless (noisy) victim transitions.

Since, the slew rate of v_i determines how fast the output transition v_o occurs, across all the circuits, we obtain different slew rates of the output transition v_o. We know that the output current has a dependence on the output voltage which is not modeled by the expression of $CGOV$ in Equation 3.1. As seen in Figure 3.5,

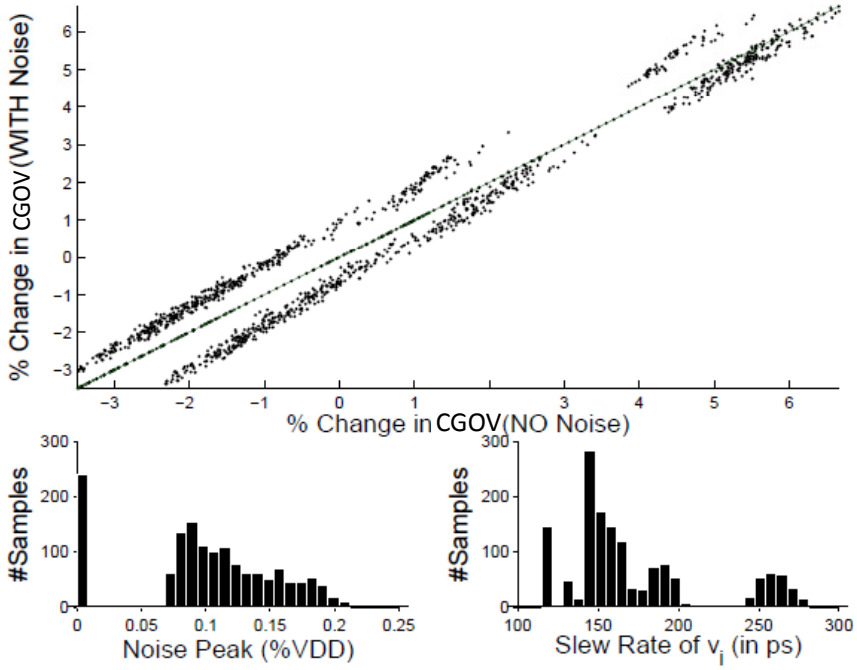


Figure 3.5: Scatter plot of *CGOV*.

the magnitude of error denoted by the spread of 10% ([-3%,7%]) around the mean *CGOV* is very small. Also, recall that the objective of computing *CGOV* is to use it for estimating the noisy victim-output arrival time. Therefore, it is necessary for the *CGOV* values of the noiseless v_i and noisy-victim transitions v_i^l to track each other very closely. It can be seen in the figure that the maximum error in the corresponding *CGOV* values across all circuits (denoted by the vertical distance from the 45^o inclined line) is only 1.23%. Hence, we conclude that the *CGOV* metric is not very sensitive to coupling noise and remains fairly constant irrespective of the shape of the input transition. This observation allows us to compute *CGOV* for the noiseless-victim transition, and use it to track the noisy victim-output arrival time.

3.3 Worst-Case Aggressor Alignment

A brute-force approach for computing the worst-case aggressor alignment would be sweeping the aggressor transition within its timing window and choosing the align-

ment that results in the latest victim-output arrival time. However, it requires multiple nonlinear simulations of the coupled aggressor-victim network and is prohibitively expensive. In this section, we show how the worst-case aggressor alignment can be computed more efficiently using the $CGOV$ metric by using only a single nonlinear simulation to obtain the coupling-noise pulse. It was also seen earlier that $CGOV$ is very robust and is fairly insensitive to the aggressor alignment. Therefore, instead of performing multiple nonlinear simulations by sweeping the aggressor transition and simulating the victim-output response, we propose to compute $CGOV$ and use it as a proxy for the victim-output arrival time.

Consider the case when both aggressor-victim drivers have a falling input transition as shown in Figure 3.6. We perform MIN analysis where we seek to find the aggressor transition that results in the earliest possible victim-output arrival time. In order to do that, we first simulate the noiseless victim input-output transitions, v_i and v_o, assuming no switching at the inputs of the aggressor driver. We then compute $CGOV$ for the noiseless-victim transition using Equation 3.1 and integrating up to the noiseless victim-output arrival time t_{out}. In the previous section, we show that $CGOV$ is fairly insensitive to the shape of the input waveform. Hence, the $CGOV$ for noisy-victim waveforms (e.g., v_i^l) can be assumed to be the same as that computed earlier for the noiseless transition v_i. Next, the noisy victim receiver-input waveform (e.g., v_i^l) is obtained with a falling transition at the input of the aggressor driver. The victim receiver-output arrival time can be estimated without actually simulating the victim-output response. The unknown victim-output arrival time (t_{out}^l) can instead be obtained by using v_i^l and integrating (using Equation 3.1) until the $CGOV$ matches that of the noiseless waveform.

Figure 3.6 illustrates the noisy-victim transitions, v_i^e and v_o^l, corresponding to the

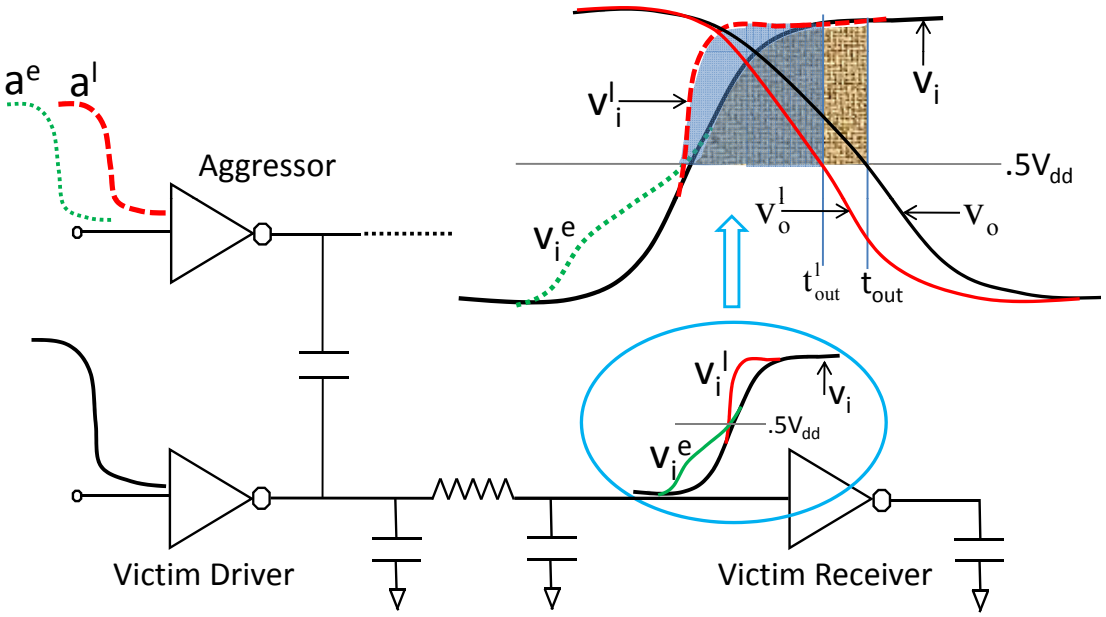


Figure 3.6: Worst-case aggressor alignment for MIN analysis.

early (a^e) and late (a^l) aggressor transitions. It can be seen that the noise aligns early for the victim transition v_i^e and does not really affect the victim waveform above the threshold voltage level, which was assumed to be 0.5VDD in this example. On the other hand, the noise due to a^l aligns later and affects the victim waveform above the threshold voltage level and causes the output to transition faster. In this example, it can be seen that the later aggressor transition (a^l) results in a faster victim-output transition v_o^l having an earlier output arrival time t_{out}^l. Finally, the worst-case aggressor alignment in MIN analysis can be chosen among all feasible aggressor transitions such that it results in the earliest arrival time at the output of the victim receiver.

A similar technique can be used in MAX analysis with mutually opposite aggressor-victim transitions. In this case, we sweep the aggressor transition within its timing window, and use the *CGOV* metric to choose the alignment which results in the

latest arrival time (t_{out}) at the output of the victim receiver. The computation complexity of the proposed approach, for every victim receiver, is $O(M * N)$, where M is the number of segments used to represent the waveforms, and N is the number of times the aggressor transition is swept within its timing window. Typically, around 10-15 segments are enough to represent the noisy waveforms fairly accurately [39]. Also, the function of the victim delay noise versus the aggressor alignment, referred to as the delay change curve (DCC) [7], is fairly well-behaved. Therefore, we can use optimization techniques (e.g., Newton-Raphson method) to reduce the number of times the aggressor alignment is swept in its timing window.

It is key to note that the proposed approach does not employ a nonlinear CSM engine to simulate the victim receiver-output response within every iteration. Instead, the proposed approach uses $CGOV$ to estimate the victim receiver-output arrival time and achieves substantial speedup over approaches which employ expensive nonlinear CSM based simulations. Further speed-up can be obtained by using a linear-superposition based framework to compute the noisy victim receiver-input waveforms corresponding to each aggressor alignment.

The proposed algorithm requires the enumeration of aggressor alignment within its timing window. With the use of optimization techniques, the number of such enumerations are typically small. Nevertheless, for every aggressor alignment, we would still need to perform expensive nonlinear simulations using the CSMs to obtain the accurate noisy-victim waveforms. Hence, to reduce the computation overhead significantly, we use the principle of linear superposition, and combine the noiseless-victim waveform with the coupling noise to obtain an estimate of the noisy victim response. It is well-known that the use of linear superposition can lead to an underestimation of the coupling-noise peak, since the change in noise peak due to the nonlinearity of

victim driver is not modeled. However, it accurately estimates the pulse width of the noise waveform. Hence, although linear superposition underestimates the noise peak, it does not necessarily underestimate the time-to-peak of the noise which is needed to estimate the alignment. Therefore, the $CGOV$ metric is not very sensitive to the nonlinearity of victim-driver resistance. In the results section, we show that the average error introduced by the superposition assumption is typically very less ($\approx 1.7\%$). Therefore, we make an engineering decision to use the principle of superposition for computing the noisy-victim waveforms. Since, nonlinear simulation using CSMs are performed only once to obtain the coupling noise and noiseless-victim waveform, the proposed alignment approach is overall very fast.

In this chapter, we analyzed the case when the victim is coupled to a single aggressor. However, in a typical circuit, the victim net is often coupled to more than a single aggressor. In such cases, it is necessary to find the worst-case alignment of all the aggressors coupled to the victim such that it results in the maximum victim-output arrival time. A heuristic often used to compute the relative alignment among aggressors is by aligning the coupling-noise pulses such that all the noise peaks coincide, which results in cumulative coupling-noise pulse having the largest noise peak. The cumulative coupling-noise pulse is then optimally aligned with the victim waveform. In contrast, any other alignment among aggressors would result in a cumulative coupling-noise pulse with a smaller noise peak and a wider pulse width. However, it was shown in [15] that using the coupling-noise pulse with the largest noise peak resulted in an error that was less than 5% across exhaustive SPICE simulations. Therefore, in this work we have focused on the alignment of the cumulative coupling-noise pulse with the victim transition such that the victim receiver-output arrival time is maximized.

Table 3.1: Parameters of the experimental aggressor-victim circuit.

Parameters	Set of Parameter Values
Aggressor Driver Strength	(2X , 12X)
Aggressor Driver Input Slew Rate (ps)	(10, 200)
Victim Driver Strength	(2X , 8X , 12X)
Victim Driver Input Slew Rate (ps)	(10, 50, 100, 200)
Victim Interconnect Length (μm)	(5, 50, 100, 200)
Coupling scaling factor k	(0.5 , 1, 1.5, 2)
Victim Receiver Load (fF)	(1 , 10 , 50 , 200)

3.4 Experimental Results

In this section, we will show experimental results that verify the accuracy and effectiveness of our proposed approach for computing the worst-case aggressor alignment. All experiments were performed on the fully-coupled aggressor-victim circuit shown in the Figure 3.7 in $65nm$ technology node. The nonlinear CMOS gates were simulated with detailed internal RC interconnects extracted from an industrial library. We assume, for the interconnect wire load model in the $65nm$ technology node [51], a wire resistance $R = 0.5\Omega/\mu m$ and a wire ground capacitance $C_g = 0.2fF/\mu m$. We know that the coupling capacitance C_c is a function of the spacing and the relative amount of overlap between the aggressor-victim nets. If the aggressor-victim nets are routed very closely in the same metal layer, then the coupling capacitance could account for a significant portion of the total wiring capacitance. Conversely, the magnitude of the coupling capacitance would be small when the aggressor-victim nets are routed at a relatively farther distance from each other. Therefore, in our experiments we obtain the coupling capacitance C_c by appropriately scaling the ground capacitance, i.e., $C_c = k * C_g$, where k is a scaling factor which ranges between $[0.5, 2]$ in our experiments.

The proposed approach of finding the worst-case aggressor alignment is validated on the aggressor-victim coupled circuit shown in Figure 3.7. However, the representa-

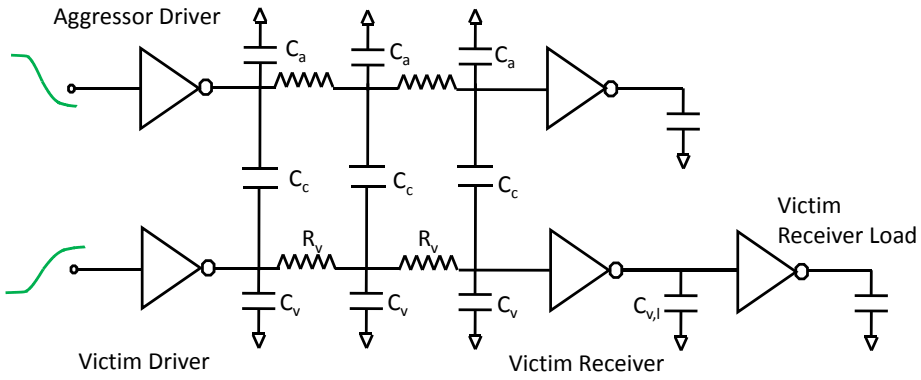


Figure 3.7: The experimental aggressor-victim circuit.

tive circuit can have several variable parameters such as the types and the strengths of the aggressor-victim drivers and receivers, the input-slew rates, the interconnect lengths, the coupling-capacitance scaling factor (k), and the receiver-output loads. In order to adequately sample the above mentioned parameter space, we perform a total of about 1500 simulations by sweeping the parameter values (as shown in Table 3.1).

In each simulation, we compare the accuracy of our proposed approach with the *golden* aggressor alignment obtained by using HSPICE based brute-force enumeration. The aggressor transition was enumerated with a discretization step size of $2ps$, and HSPICE simulations were performed to simulate the victim receiver-output response for every aggressor alignment. Finally the aggressor transition that results in the latest victim-output arrival time was reported as the worst-case aggressor transition. In comparison, the proposed approach computed the worst-case aggressor alignment by using the $CGOV$ metric to predict the victim-output arrival time. Once, the aggressor alignment is computed, HSPICE was used to simulate the victim receiver-output response. In order to evaluate the accuracy, we express the difference in the victim receiver-output arrival time with respect to golden as a percentage of

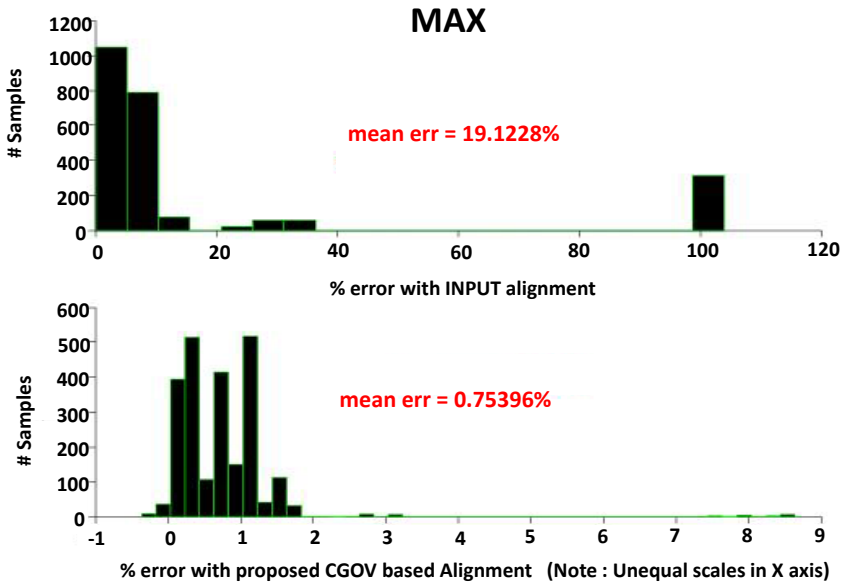


Figure 3.8: Histogram of %Error in delay noise for MAX analysis.

the worst-case delay noise at the output of the victim receiver. A histogram of the percentage error in delay noise with the CGOV based alignment, obtained across all simulations, is shown in Figure 3.8 (3.9) for MAX (MIN) analysis, when the victim-receiver gate is an inverter with a falling-input transition. It can be seen that the proposed approach accurately estimates the worst-case aggressor alignment, and we observe an average error of 0.75% (2.1%) for the MAX (MIN) analysis.

Earlier, we claimed that aggressor alignment that maximizes the victim receiver-*input* arrival time does not necessarily maximize the victim receiver-*output* arrival time. In order to validate the above claim, we perform a similar brute-force enumeration, and compute an aggressor alignment such that it maximizes the victim receiver-input arrival time. Using the above computed aggressor alignment, we run HSPICE to simulate the victim receiver-output transition. Finally, we compare the percentage error in the output arrival time with respect to golden, and plot the histogram of the error across all the simulations (Input alignment in Figures 3.8-3.9).

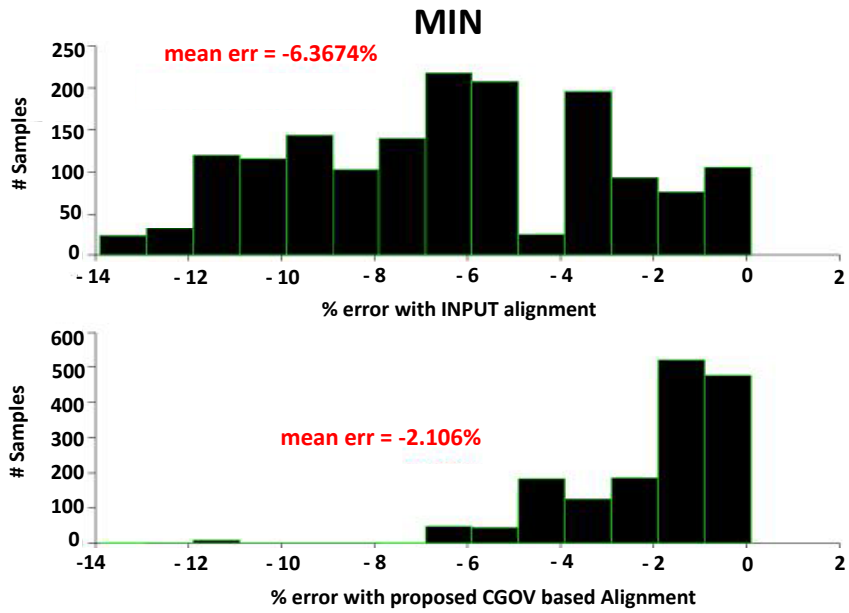


Figure 3.9: Histogram of %Error in delay noise for MIN analysis.

It can be seen that more than 10% of the cases report an error of 100% in delay noise for MAX analysis. These cases occur when the input alignment results in a coupling noise which aligns too late with the victim transition (as shown in Figure 3.1). Overall, it can be seen that the proposed approach performs better than the input-alignment approach, and establishes the significance of considering the victim-receiver gate in the computation of the worst-case aggressor alignment.

We repeat the above experiment for different victim-receiver gates, and in Table 3.2 we show the average percentage error in delay noise with respect to golden. It can be seen that the proposed approach accurately computes the worst-case aggressor alignment across all the different receiver gates. It is interesting to note that for the rising victim receiver-input transition (RISE MAX), the maximum error (3.85%) in delay-noise estimation occurs for the four-input NOR receiver gate. Similarly, the proposed approach results in a mean error of 4.41% for the four-input NAND receiver gate. The increase in error could be due to the *stack effect* of the pull-up

Table 3.2: Mean %Error in delay noise of the victim compared to golden analysis.

	RISE MAX		FALL MAX		RISE MIN		FALL MIN	
Receiver	CGOV	Inp Align	CGOV	Inp Align	CGOV	Inp Align	CGOV	Inp Align
INV	0.77	7.24	0.75	19.12	-2.32	-6.66	-2.10	-6.36
NAND4X	0.35	7.34	4.41	5.11	-2.99	-12.87	-3.07	-4.50
NOR4X	3.85	5.36	1.38	8.36	-3.35	-7.39	-2.23	-12.31
BUF	0.26	8.34	0.31	8.42	-1.67	-7.20	-1.34	-9.87
AND4X	0.29	8.36	1.86	8.19	-1.51	-6.14	-1.07	-8.75
OR4X	1.46	7.63	0.37	9.92	-0.99	-6.06	-2.23	-12.40

(pull-down) networks for the NOR (NAND) gates which affects the robustness of the $CGOV$ metric used to track the victim-output arrival time. However, one can note that even with the stack effect, the proposed approach is more accurate than the input-alignment approach. Across all simulations, the average error in delay noise for the proposed approach is 1.7% compared to an error of 8.49% obtained with the input-alignment approach.

3.5 Summary

In this chapter, it was seen that worst-case aggressor alignment must be computed such that it always maximizes the victim receiver-output arrival time. In order to model the victim receiver-output waveform, we define and use the cumulative gate overdrive voltage ($CGOV$) metric to model the total victim receiver-output current. Since, the victim receiver-output transition directly depends on the amount of current sourced by the receiver gate, the alignment with the lowest $CGOV$ would correspondingly lead to the slowest transition having the maximum delay. HSPICE simulations, performed on industrial nets to validate the proposed methodology, show an average relative error of 1.7% in delay noise compared to the worst-case alignment obtained by performing an exhaustive sweep.

CHAPTER IV

Top-k Aggressors in Noise Analysis

Due to the aggressive scaling of interconnect wires, capacitive-coupling noise has become an important concern for nanometer designs. Therefore, delay noise which models the impact of coupling noise on the circuit delay is of particular concern for high-performance designs. To address this issue, noise analysis was first introduced in [65, 66] and has been the focus of significant research effort. Since delay noise requires both aggressor-victim nets to switch at the almost the same time, timing windows were defined to indicate the time range in a clock period within which the aggressor-victim nets can transition. It was observed that the computation of delay noise and timing windows poses a chicken-and-egg problem. Delay noise cannot be computed before timing windows are defined and, vice versa, accurate timing windows cannot be computed without information about the delay noise. An iterative method was suggested in [61, 37] for computing the delay noise. The iterations start by either assuming that all aggressor-victim timing windows have an overlap or none of them have any overlap. The circuit delay is then computed by iteratively updating the delay noise and timing windows. It was shown in [61] that this iterative method is guaranteed to converge. It was observed in [77] that the above problem has solutions which are fixpoints on a complete lattice.

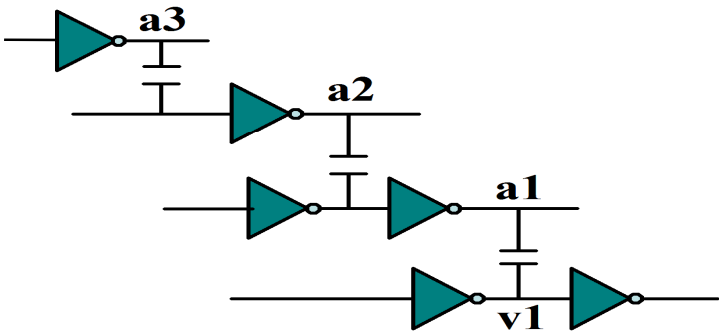


Figure 4.1: Indirect aggressors a2, a3 affect the timing window of the primary aggressor a1 in the noise analysis for victim v1.

A number of methods to identify and eliminate false aggressors, which cannot impact the delay of a victim due to logical and timing correlations in the circuit, have been proposed in [10, 58, 27, 14, 18]. Despite the pruning of false aggressors, the total number of aggressors that contribute to the delay noise of the circuit can be very high. On every victim net, the primary aggressors couple noise directly on the victim transition. Furthermore, indirect aggressors that are coupled to primary aggressors can impact the timing window of a primary aggressor. The increase in the timing window of a primary aggressor can lead to an overlap with the victim timing window, resulting in an increase in the amount of delay noise on the victim. In Figure 4.1, the noise coupled from aggressor *a2* can increase the timing window of *a1* such that it overlaps with that of victim *v1*. Therefore, *a2* is an indirect or secondary aggressor of victim *v1*. Similarly, *a3* is a tertiary aggressor of victim *v1*. Note that in this example, noise analysis would require three iterations for convergence. The fact that industrial noise-analysis tools report the need for three-four iterations for convergence [49], shows that the noise from indirect aggressors contributes to the circuit delay noise in industrial designs. Since, we must consider all primary aggressors coupled to the critical path and also indirect aggressors coupled to the

transitive-fanin cone of the primary aggressors, the number of aggressors that can potentially contribute to the circuit delay is huge. However in practice, designers often limit the number of aggressors that can switch simultaneously due to one of the following reasons:

- Delay noise that involves hundreds of precisely-timed noise events is considered unlikely and consequently ignored.
- A noise event involving hundreds of aggressors is less probable than that involving a few aggressors.

A very common approach to limit the total number of aggressors considered in noise analysis is by restricting the set of primary aggressors for each victim to a few (say ten) by choosing only those aggressors which cause a significant amount of coupling noise on the victim. However, this approach of reducing the number of aggressors is unsystematic and may lead to unpredictable results. The total number of aggressors that contribute to path delay noise will vary from one path to another. Also, there is no consistent manner of restricting the total number of indirect aggressors that contribute to delay noise due to noise-analysis iterations. With limited resources for fixing delay-noise violations, designers would prefer to fix a fewer number of critical aggressors. In this chapter, we present an approach for computing the set of *top-k* aggressors that contributes strongly to the circuit delay noise. The concept of the set of *top-k* aggressors is analogous to the *top-k* critical paths that is commonly reported in traditional STA, but comes in two flavors,

Top-k aggressors' elimination set: Given traditional noise analysis, the *top-k* aggressors' elimination set is a set of k aggressor-victim couplings which, when not considered in noise analysis, would reduce the delay noise of the circuit by a maximum

amount. This information is vital to a circuit designer in situations where only a limited number of aggressor-victim couplings can be fixed. Suppose a circuit designer can eliminate only ten coupling capacitances, for instance, through wire shielding or spacing. Then the top-10 aggressors' elimination set reports the set of ten aggressor-victim couplings which must be fixed to obtain the maximum reduction in delay noise. Hence, the *top-k* aggressors' elimination set ensures that the maximum improvement in delay noise is achieved for the performed effort. It is true that fixing a particular aggressor-victim coupling may perturb the overall physical implementation and may result in other new couplings. Nevertheless, the availability of the *top-k* aggressors' elimination set is key in each cycle of delay-noise mitigation.

Top-k aggressors' addition set: Given a timing analysis without considering delay noise, the *top-k* aggressors' addition set is the set of k aggressor-victim couplings whose delay noise, when added to the noiseless timing analysis, will result in the maximum circuit delay. The *top-k* aggressors' addition set is useful when the designer wants to restrict the noise analysis to no more than k aggressor-victim couplings switching together. Alternatively, it can also be used to identify the sets of aggressors which must be given a higher priority while fixing aggressors for delay-noise mitigation.

We show that the computation of the *top-k* aggressors' addition and elimination sets are dual problems. The analysis is complicated by the fact that both primary and indirect aggressors must be considered for inclusion in the *top-k* aggressor sets. Furthermore, we must efficiently model the propagation of delay noise in the circuit. The proposed algorithm uses two novel concepts to provide a tractable solution: Firstly, we model the propagated delay noise with a pseudo-input aggressor, and secondly, we prune the enumeration space by using a dominance relationship which

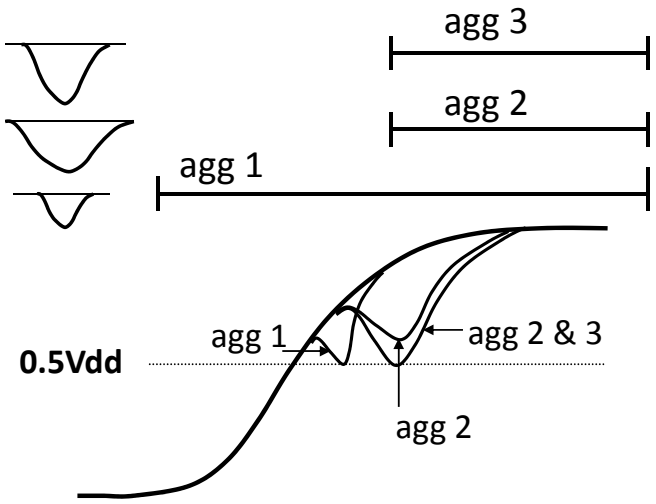
imposes a partial ordering on the aggressor sets. The proposed algorithm is able to achieve a practical runtime for large values of k on all benchmark circuits. In comparison, brute-force enumeration could not generate aggressor sets with k greater than three, even for the smallest benchmark circuit. The remainder of this chapter is organized as follows. We describe the problem in detail in Section 4.1 and then describe the proposed algorithm for computing the *top-k* aggressors' set in Section 4.2. Experimental results are shown in Section 4.3, and we summarize the chapter in Section 4.4.

4.1 Problem Description

The goal of this work is to identify, for a given k, the set of k aggressors which must be fixed (refered to as *top-k* aggressors' set) for minimizing the delay-noise violations in a design. This work can (potentially) be employed in the inner loop of circuit optimization and therefore runtime efficiency is key. Conventionally, linear-driver models were used to perform noise analysis efficiently. Recently, nonlinear current-source based driver models have been proposed [24] to achieve the accuracy demanded by timing analysis. However, a single aggressor-victim alignment in such a framework requires a nonlinear solver, and it is difficult to achieve an efficient runtime. Therefore, in this work, we make an engineering decision to use linear-driver models, since they are still used in the industry [49] in applications (such as ours) where accuracy can be traded for runtime efficiency.

4.1.1 Challenges in Computing the Top-k Aggressors' Set

The worst-case aggressor-victim alignment problem has been extensively studied in literature [37, 75, 57]. However, the problem of computing the set of *top-k* aggressors has not been addressed, to our knowledge. A brute-force manner of generating



k	Top-k set
1	agg {1}
2	agg's {2, 3}

Figure 4.2: Non-monotonicity in aggressor lists for aggressors a1, a2 and a3.

the *top-k* aggressors' elimination set is by simply performing noise analysis multiple times and eliminating k aggressors in each run. In this case, a total of ${}^{r}C_{k}$ noise-analysis runs are required, where r refers to the total number of aggressors in the circuit and is very expensive computationally. A bottom-up construction of the *top-k* aggressors' set is nontrivial and complicated due to several factors and two of which are discussed below.

First, with increasing cardinality, the sets of *top-k* aggressors could be nonmonotonic. For example, suppose the *top-3* aggressors' set in an arbitrary circuit contains aggressors {*a1*, *a2*, *a3*}. Then, the set of aggressors with the next higher cardinality (i.e., *top-4*) may not necessarily contain any of these aggressors. This property arises due to the fact that the alignment of aggressors affects the delay noise, and we illustrate the above with an example. Consider a victim net coupled to three aggressors {*a1*, *a2*, *a3*} with timing windows as shown in Figure 4.2. Although the aggressors *a2* and *a3* have larger coupling-noise pulses than *a1*, their timing windows restrict their alignment to the left, and they do not cause any delay noise on the victim

when they switch individually as the victim t_{50} does not change. Hence, the *top-1* aggressor set is {*a1*} despite the fact that its coupling-noise pulse is smaller than that of both *a2* and *a3*. However, when we consider the *top-2* aggressors' set, the delay noise due to the simultaneous switching of aggressors {*a2*, *a3*} is greater than that of {*a1*, *a2*} and {*a1*, *a3*}. Therefore, the *top-2* aggressors' set is {*a2*, *a3*}. This example shows that adding an aggressor to the *top-k* aggressors' set may not necessarily produce the *top-*{*k+1*} set.

Secondly, the worst-case aggressor-victim alignment at a victim net is affected by the delay noise propagated from the transitive-fanin cone of the victim driver. Hence, the *top-k* aggressors' set at any victim must consider aggressors coupled to the transitive-fanin cone of the victim. Similarly, the impact of aggressors coupled to the transitive-fanin cone of the primary aggressors must also be considered as they can change the timing window of the primary aggressors. Therefore, the primary aggressors must be considered both by themselves as well as acting in concert with tertiary aggressors that increase their timing window. A primary aggressor acting by itself is referred to as a first-order aggressor. Primary aggressors are assigned an order $p = t+1$, where t is the number of aggressors coupled to the transitive-fanin cone of the primary aggressor.

4.2 Proposed Approach

We will focus on the algorithm to compute the *top-k aggressors' addition set*, since it is conceptually simpler to understand. Later in Section 4.2.4, we show how the proposed algorithm can be modified to instead compute the *top-k aggressors' elimination set*. The desired set of *top-k* aggressors is iteratively computed in a bottom-up manner using implicit enumeration. During the i^{th} iteration ($i < k$),

a super-set of all aggressor sets of cardinality i (defined as $List_i$) is constructed. Elements of $List_i$ can potentially be a subset of the desired set of *top-k* aggressors, and the $List_i$ is generated for all values of i from one through k. Finally, the set of aggressors in $List_k$ which results in the maximum delay noise is reported as the set of *top-k* aggressors. The enumeration of aggressor sets in $List_i$ (in the i^{th} iteration) is based on two key concepts:

First, we use *pseudo-input aggressors* to model the shift in a victim transition due to the coupling noise on the transitive-fanin cone of the victim driver. This allows us to model all aggressors coupled to the transitive-fanin cone of a victim by using pseudo-input noise envelopes that are similar to the noise envelopes of primary aggressors. Hence, if $List_i$ is computed at the input of a victim driver, then the $List_i$ can be propagated to the victim net by using these pseudo-input aggressors. Hence, we can efficiently partition the problem by propagating $List_i$ to every node of the circuit in a topological order.

Second, we use the concept of dominance to aggressively prune the solution search space. The dominance property imposes a partial ordering on the aggressors of a victim. If the noise envelope *N1* of aggressor *a1* entirely encompasses the noise envelope *N2* of another aggressor *a2*, then the delay noise due to *N1* is never less than that of *N2*. Therefore, while computing the *top-1* aggressor set, aggressor *a1* will be chosen as compared to *a2*. In other words, aggressor *a1* dominates aggressor *a2*. This dominance property can easily be extended to pseudo-input aggressors and higher-order aggressors. Note that dominance-based pruning dramatically improves runtime for large values of k. We will describe the concepts of pseudo-input aggressors and dominance in more detail, and then present the algorithm to compute the set of *top-k* aggressors.

4.2.1 Pseudo-Input Aggressors

Delay noise propagated from the input of a victim driver affects the alignment of the downstream victim nets with their respective primary aggressors. Therefore, the set of *top-k* aggressors for a victim net may comprise of the primary aggressors and the aggressors coupled to its fanin-cone. A brute-force approach to compute the set of *top-k* aggressors for a victim would be

- Select p ($p < k$) aggressors coupled to the transitive-fanin cone of the victim, and compute the delay noise that is propagated to the victim.
- Select k - p primary aggressors coupled to the victim net. Compute the worst-case alignment among aggressors and the corresponding delay noise on the victim.
- Enumerate all possible combinations of aggressor sets for all values of p and repeat the delay-noise calculations. Finally, we select the set of aggressors that produces the maximum delay noise on the victim.

Clearly, the brute-force method is prohibitively expensive as there can be numerous sets of fanin aggressors and cannot be employed for any large circuit. In order to make the problem tractable, we introduce the concept of pseudo-input aggressors which allows us to propagate the *top-k* aggressor sets in a topological order.

The noiseless and noisy input-output transitions of a typical victim net are shown in Figure 4.3. We know that a noisy victim-input transition leads to a noisy victim-output transition that can alter the worst-case alignment of the victim with its primary aggressors. In order to propagate the *top-k* aggressors' sets in a topological order, we wish to break the dependence of the concerned victim-output transition on the noisy input transition at its fanin. We represent the coupling noise

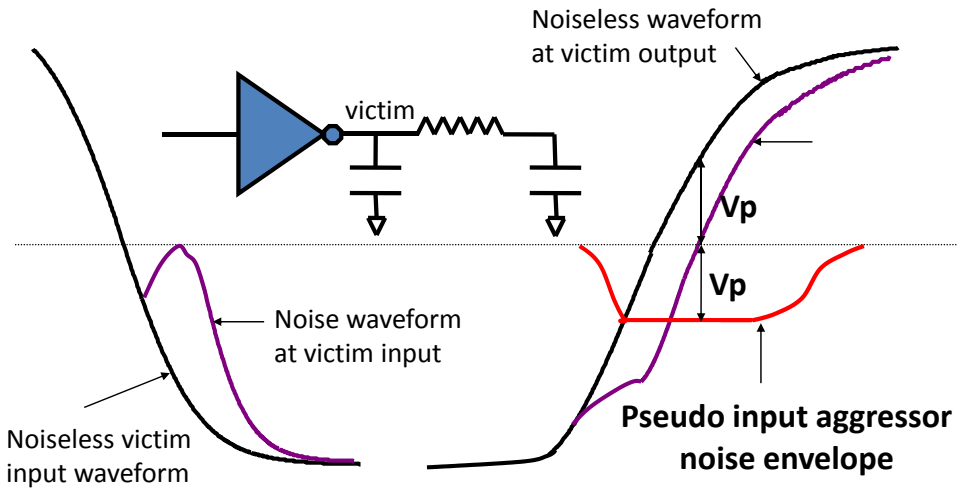


Figure 4.3: Pseudo-input noise envelope.

contributed from the worst-case set of aggressors coupled to the fanin of the victim with a pseudo-input noise envelope. A pseudo-input noise envelope is defined as the waveform obtained by subtracting the noiseless-victim transition from the noisy-victim transition (see Figure 4.3). Using the principle of linear superposition, the noisy-victim transition can be constructed by appropriately superimposing this pseudo-input noise envelope with the original noiseless-victim transition. Note that this pseudo-input noise envelope has a shape that is similar to the noise envelope obtained from primary aggressors. Due to the circuit layout, the nets present in the topological fanout of a victim net could become its primary aggressors. The delay noise induced by such aggressors can be modeled accurately by their pseudo-input aggressors.

4.2.2 Aggressor Dominance

During the bottom-up enumeration procedure, in the i^{th} iteration ($i > 1$), we have several candidate aggressor sets within $List_i$, which are potential subsets of the desired set of *top-k* aggressors. In a naive implementation, the $List_i$ can easily blow

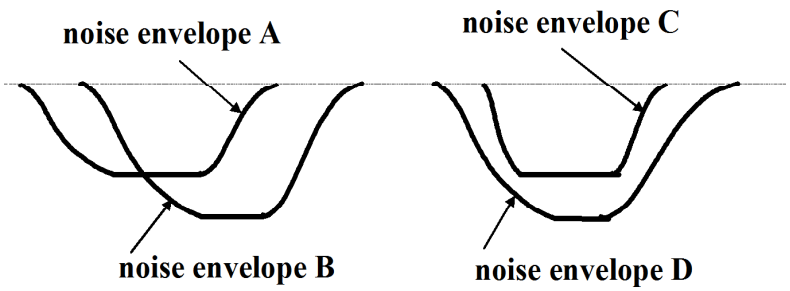


Figure 4.4: Dominance between the noise envelopes.

up and it is important to keep the cardinality of $List_i$ under check, which is achieved by using the notion of dominance.

- **Dominance:** For any victim, if the noise envelope of aggressor A encapsulates that of any aggressor B, then aggressor A is said to dominate aggressor B.

As shown in Figure 4.4, the noise envelope D dominates the noise envelope C, whereas neither A or B are mutually dominated. The usefulness of dominance follows from the following theorem,

Theorem 4.1. *Consider two aggressors' sets* P *and* Q *with the same cardinality such that* P ***dominates*** Q*. A higher-cardinality set of aggressors obtained by adding any additional aggressor* $\boldsymbol{a}$ *to* Q *would never couple a greater delay noise on the victim than that obtained by adding* $\boldsymbol{a}$ *to* P*.*

Proof. Given P dominates Q, the noise envelope of P must encapsulate that of Q. Now, the cumulative-noise envelope $P \cup \mathbf{a}$, obtained by adding the noise envelopes of P and $\mathbf{a}$, must at each point in time either encapsulate or be equal to that of the cumulative-noise envelope $Q \cup \mathbf{a}$. As the magnitude of noise of $Q \cup \mathbf{a}$ is never greater that of $P \cup \mathbf{a}$, the noisy-victim transition obtained by superimposing $Q \cup \mathbf{a}$ with the noiseless-victim transition will always have a higher (lower) voltage for a rising (falling) victim transition than $P \cup \mathbf{a}$. Therefore, the noisy victim t_{50} of $Q \cup$

a must always be earlier than that of $P \cup \mathbf{a}$. Consequently, the delay noise of $P \cup \mathbf{a}$ is always larger than that of $Q \cup \mathbf{a}$. □

Consequently, we do not need to propagate dominated aggressor such as aggressor C (as shown in Figure 4.4), since we can always replace it by aggressor D which produces a higher delay noise. This observation naturally leads to the creation of irredundant lists, defined as:

- **Irredundant Lists:** If $List_i$ be the list of all possible aggressor sets each having cardinality i, then the Irredundant List ($I\text{-}List_i$) is a subset of $List_i$, such that all aggressor sets $x \in I\text{-}List_i$ are not dominated by any aggressor set $y \in List_i$.

In other words, $I\text{-}List_i$ consists of all sets of non-dominated aggressors of cardinality i. The fact that the desired set of *top-k* aggressors is a subset of $I\text{-}List_k$, reduces our search space significantly.

Finally, we show how the dominance property can be applied to the aggressor noise envelopes. We first identify a time interval, referred to as the dominance interval, within which a noise envelope has to encapsulate another noise envelope for the dominance relationship to hold. The lower bound of the dominance interval is the t_{50} of noiseless victim, since a noise envelope that ends before the t_{50} will not induce any delay noise. For the other boundary of the dominance interval, we compute an upper bound on the delay noise by performing standard noise-analysis by assuming all aggressors to have infinite timing windows. In practice, we find that a large number of noise envelopes dominate each other within the dominance interval, thereby reducing the search space significantly.

4.2.3 Algorithm to Compute the *Top-k* Aggressors' Addition Set

In this subsection, we explain the algorithm for computing the *top-k* aggressors' addition set. For a desired value of k, the goal of the algorithm is to construct the irredundant lists $I\text{-}List_k$ at the sink node by performing implicit enumeration. In the i^{th} enumeration step of the algorithm, we compute the corresponding $I\text{-}List_i$ by operating on the $I\text{-}List_{i-1}$ computed in the previous step. Using the concept of pseudo-input aggressors and Theorem 4.1, we can propagate the irredundant lists through the circuit in topological order to obtain the desired irredundant list at the sink node of the circuit. The *top-k* aggressors' addition set is the set which belongs to $I\text{-}List_k$ and produces the maximum delay noise. The pseudo-code of the proposed algorithm to compute the *top-k* aggressors addition set is given below.

COMPUTE TOP-K AGGRESSORS' ADDITION SET(k)

1 **for** $i \leftarrow 1$ **to** k
2 **for** *every victim net* (in topological order)
3 Create $List_i$ by {
4 Adding a non-dominated primary aggressor to each set of $I\text{-}List_{i-1}$;
5 Adding a pseudo-input aggressor of cardinality i ; (For multiple-input gates, pick the one resulting in the latest victim-output arrival time)
6 Adding non-dominated higher-order aggressors of cardinality i ;
7 }
8 $I\text{-}List_i \leftarrow List_i$ (*using Theorem 4.1*) ;
9 Find the aggressor set $agg_{wc} \in I\text{-}List_i$ having the maximum delay noise ;
10 Propagate agg_{wc} as the pseudo-input aggressor of cardinality i ;
11 Return the set of *top-k* aggressors $agg_{wc,sink} \in I\text{-}List_k$ of the sink node ;

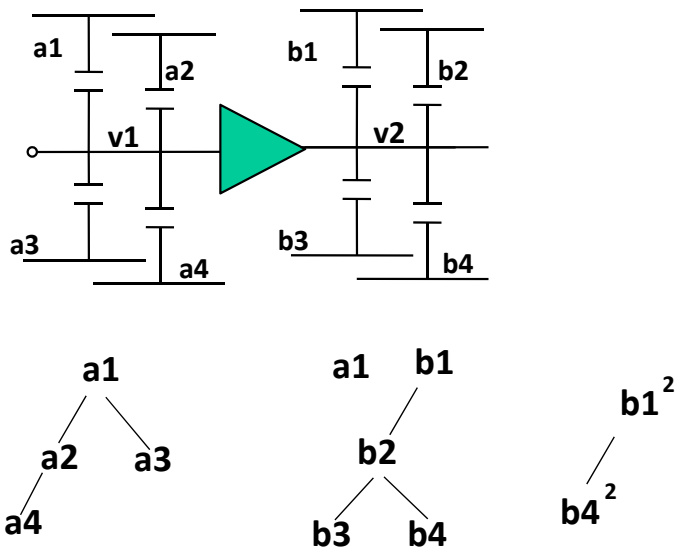


Figure 4.5: Partial ordering between noise envelopes.

For ease of understanding, we explain the steps of algorithm in more detail using an example (as shown in Figure 4.5). In the Figure 4.5, we have two victim nets *v1* and *v2*, with *v1* being the input to the driver of *v2*. Victim *v1* is coupled to four primary aggressors *a1-a4*, and similarly *v2* is coupled to aggressors *b1-b4*. Running traditional noise-analysis, we obtain the timing windows and the noise envelopes of all the primary aggressors on each victim net. The partial ordering on the noise envelopes of the aggressors based on the dominance is also shown in Figure 4.5, where aggressor *a1* dominates all the other primary aggressors (i.e., *a2, a3, a4*), and *b1* dominates all primary aggressors (i.e., *b2, b3, b4*). In the first step, we would like to find all non-dominated aggressors of cardinality one. Since *v1* is a primary input, it has no pseudo-input aggressors. The irredundant list of cardinality one ($I\text{-}List_1$) has one set of aggressors $\{(a1)\}$ (as shown in Table 4.1).

For victim *v2*, we propagate only aggressor *a1* as a pseudo-input aggressor, since it is the only aggressor present in $I\text{-}List_1$ for *v1*. If the victim driver had multiple inputs, then we would compute the set of *top-k* aggressors for each input independently, and

Table 4.1: Creating higher-order irredundant sets.

	Irredundant list for v1	Aggressor sets for v2	Irredundant list for v2
k=1	$(a1)$	$(a1)\,(b1)$	$(a1)\,(b1)$
k=2	$(a1,a2)\ (a1,a3)$	$(a1,b1)(b1,b2)$ $(a1,a2)\ (a1,a3)\ \left(b1^2\right)$	$(a1,b1)(b1,b2)$ $(a1,a3)\ \left(b1^2\right)$
k=3	$(a1,a2,a3)(a1,a2,a4)$	$(b1,b2,b3)(b1,b2,b4)$ $(a1,a3,b1)\left(b1^2,a1\right)$ $(a1,a2,a3)(a1,a2,a4)$	$(b1,b2,b3)(a1,a3,b1)$ $\left(b1^2,a1\right)(a1,a2,a3)$

select the one resulting in the maximum victim-output arrival time. In the example, the pseudo-input aggressor *a1* is not dominated by aggressors *b1-b4*. Hence, $I\text{-}List_1$ for victim *v2* includes two aggressor sets $\{(a1),\ (b1)\}$.

Now, $I\text{-}List_2$ for victim $v1$ can be computed by the explicit enumeration of all possible pairs of aggressors coupled to $v1$ and its transitive-fanin cone. However, a more effective approach is to reuse the information from the previous irredundant list (i.e., $I\text{-}List_1$). In the example, both sets (*a1,a2*) and (*a1,a3*) are added to the $I\text{-}List_2$ for the victim *v1*. Using Theorem 4.1, we can ignore dominated sets such as (*a2, a4*). $I\text{-}List_2$ for victim *v2* is computed by first adding aggressor *b2* to each set in $I\text{-}List_1$. Next, we add the pseudo-input aggressors of cardinality two. The set of non-dominated aggressors propagated from *v1* is {(*a1,a2*), (*a1,a3*)}. Finally, we account for the aggressors having an order two (i.e., whose timing window has increased due to one additional aggressor, e.g., b_1^2). Note that the noise-envelope height of an aggressor with order two is the same as its order one counterpart. However, the width of the noise envelope increases as it has a larger timing window due to delay noise. In this example, using the property of dominance, we find that b_1^2 dominates every other order two aggressor and is added to $I\text{-}List_2$. Note that at this point, $I\text{-}List_2$ for *v2* contains five entries (see Table 4.1). However, some of these entries may dominate each other and can therefore be reduced further as shows in the right-most column of Table 4.1. Similarly, we generate the $I\text{-}List_3$ by

operating on $I\text{-}List_2$. Similarly, we repeat the procedure for k iterations to finally obtain the $I\text{-}List_k$ at the sink node. We then superpose the noise envelopes from all the aggressor sets present in $I\text{-}List_k$ of the sink node with the latest occurring noiseless-victim transition at the sink node. The set of *top-k* aggressors is the one that belongs to $I\text{-}List_k$ of the sink and results in the worst-case delay noise.

4.2.4 Algorithm to Compute the *Top-k* Aggressors' Elimination Set

In this section, we explain how the analysis for finding the *top-k* aggressors' addition set can be easily modified to find the *top-k* aggressors' elimination set. For the latter, we first assume that all aggressors are present in the circuit couple noise, and we wish to find the set of *top-k* aggressors such that fixing them will reduce the delay noise by a maximum amount. The key difference in both algorithms is that for the *top-k* aggressors' addition set, we start with noiseless timing windows and for the the *top-k* aggressors' elimination set, we start with noisy timing windows. Therefore, the noise envelopes of all primary aggressors are expanded as their timing windows account for delay noise. The dominance property remains the same, and irredundant lists of aggressor sets are computed in a similar manner. However, when superposing the noise envelopes, we want to reduce the delay noise in the design. Hence, we first define a total noise envelope obtained by adding up the noise envelopes from all the aggressors with their largest timing windows, such that it results in the maximum delay noise in the circuit. The superposition of the noise envelopes from the aggressor sets can be performed as follows: (1) subtract the noise envelope from the total noise envelope, (2) superpose the resulting envelope with the noiseless-victim transition, and (3) select the set that results in the smallest delay noise. The overall algorithm functions in a similar manner and the *top-k* aggressors' elimination set is selected from the I-$List_k$ of the sink node such that it results in the minimum circuit delay.

4.3 Experimental Results

In this section, we show experimental results of the proposed *top-k* aggressors' addition and elimination algorithm by using a prototype noise-analysis tool implemented in C++. A 130nm standard-cell library was used for synthesis and technology mapping of the LGSynth91 benchmark circuits [1], whose circuit sizes vary between 50-3000 gates. The Cadence Silicon Ensemble tool was used to perform the place-and-route with a target utilization of 80% and using up to three metal layers. The Mentor Graphics Calibre tool was used to extract the parasitic coupling and ground capacitances for the nets in the design.

A brute-force algorithm, as explained earlier in Section 4.2.1, was also implemented to verify the accuracy of the proposed algorithm. However, the enormous complexity of the brute-force algorithm resulted in its failure to generate the *top-k* aggressors' sets for $k > 3$ in *1800* seconds for the even smallest circuit $i1$. In comparision, the proposed algorithm was able to generate the set of *top-50* aggressors with a tractable runtime for all benchmark circuits. As shown in Table 4.2, for values of $k < 3$, the set of *top-k* aggressors computed by the algorithm was consistent with that obtained by the brute-force method. Note that the proposed enumeration approach leads to a significant speed-up over the brute-force approach.

Figure 4.6 plots the circuit delay of the benchmark circuit $i10$ as a function of the

Table 4.2: Validation of the proposed approach with brute-force enumeration for circuit i1.

	Brute-Force Enumeration		Top-k Elimination set	
k	ckt delay (ns)	runtime (s)	ckt delay (ns)	runtime (s)
1	0.743	0.12	0.743	0.01
2	0.722	9.65	0.722	0.01
3	0.709	621.4	0.709	0.02
4	-	-	0.703	0.06

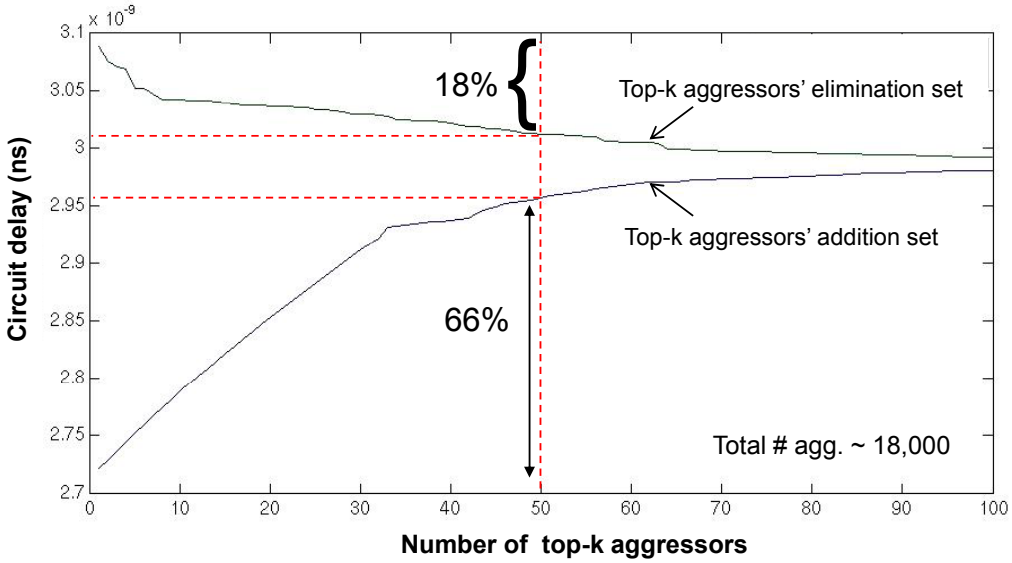


Figure 4.6: Plot of circuit delay versus k for circuit i10.

number of *top-k* aggressors for both the *top-k* aggressors' addition and elimination sets. It can be seen that the top-50 aggressors' addition set accounts for about 66% of the circuit delay noise. Also, out of a total of 18,000 aggressors, fixing those aggressors that belong the top-50 aggressors' elimination set leads to about 18%

Table 4.3: Delay and runtime results for computing the top-k aggressors' addition set.

	Circuit characteristics			Circuit Delay (in ns)							Runtime (in s)						
ckt	# gates	# nets	# caps	no agg.	k= 5	k= 10	k= 20	k= 30	k= 40	k= 50	no agg.	k= 5	k= 10	k= 20	k= 30	k= 40	k= 50
i1	59	46	232	.452	.466	.480	.499	.520	.527	.534	.01	.01	.01	.06	.32	.65	.89
i2	222	221	706	.582	.604	.636	.667	.696	.711	.726	.01	.05	.15	.48	.81	1.44	1.68
i3	132	126	551	.413	.428	.444	.459	.489	.504	.521	.01	.02	.08	.17	.46	.73	1.12
i4	236	230	1181	.661	.674	.689	.716	.743	.764	.779	.04	.13	.17	.92	1.82	3.64	6.78
i5	204	138	1835	.958	.984	1.01	1.03	1.06	1.08	1.11	.02	.26	.82	2.52	6.86	13.2	15.4
i6	735	668	7298	.861	.898	.924	.960	.971	1.01	1.03	.09	.72	2.36	3.57	4.12	26.1	38.4
i7	937	870	9605	.823	.843	.862	.898	.932	.964	.993	.15	.61	4.12	13.9	19.5	41.8	68.1
i8	1609	1528	10235	1.47	1.50	1.52	1.54	1.57	1.58	1.59	.21	.67	2.37	9.42	16.3	37.1	66.9
i9	1018	955	14140	1.52	1.56	1.59	1.65	1.71	1.75	1.78	.18	.68	3.17	12.1	24.9	43.9	75.6
i10	3379	3155	18318	2.71	2.75	2.78	2.85	2.91	2.94	2.96	.46	.78	4.28	13.8	27.5	55.6	81.5

Table 4.4: Delay and runtime results for computing the top-k aggressors' elimination set.

	Circuit characteristics			Circuit Delay (in ns)								Runtime (in s)						
ckt	# gates	# nets	# caps	all agg.	k= 5	k= 10	k= 20	k= 30	k= 40	k= 50	no agg.	all agg.	k= 5	k= 10	k= 20	k= 30	k= 40	k= 50
i1	59	46	232	.546	.521	.513	.489	.465	.456	.453	.452	.01	.01	.02	.15	.49	.79	1.12
i2	222	221	706	.743	.695	.671	.649	.633	.625	.621	.582	.03	.07	.20	.71	1.38	2.47	3.13
i3	132	126	551	.529	.471	.453	.438	.431	.427	.421	.413	.03	.07	.10	.29	.71	1.24	1.93
i4	236	230	1181	.801	.763	.746	.716	.712	.702	.697	.661	.04	.15	.21	1.15	2.47	4.19	7.63
i5	204	138	1835	1.21	1.11	1.09	1.04	1.01	1.00	1.00	.958	.04	.46	.97	3.42	7.31	13.2	18.7
i6	735	668	7298	1.05	.976	.960	.955	.949	.936	.921	.861	.16	.62	2.2	4.29	12.5	28.5	42.5
i7	937	870	9605	1.12	1.09	1.06	1.05	1.04	1.04	1.02	.823	.20	.69	4.2	14.8	25.4	47.2	71.5
i8	1609	1528	10235	1.64	1.61	1.59	1.58	1.57	1.55	1.54	1.47	.24	.72	3.6	12.5	21.7	42.7	79.3
i9	1018	955	14140	1.84	1.81	1.80	1.78	1.77	1.76	1.75	1.52	.31	.78	3.6	13.4	26.4	41.8	75.9
i10	3379	3155	18318	3.09	3.05	3.04	3.03	3.03	3.02	3.02	2.71	.41	1.2	5.4	16.4	34.6	62.8	91.4

reduction in the circuit delay noise.

The circuit delay and runtime results obtained as a function of k for both algorithms are shown in Tables 4.3-4.4. Although the worst-case complexity of the proposed algorithm is exponential, in practice due to the efficient pruning of search space, the runtime of the algorithm grows at a smaller rate. In fact, the set of top-50 aggressors is computed for all benchmark circuits in less than 100 sec.

4.4 Summary

In this work, we introduced the concept of *top-k* aggressors for fixing noise violations. The problem of computing the *top-k* aggressors' set is nontrivial and a naive brute-force implementation leads to impractical runtime. We proposed the concept of a pseudo-input aggressor that allows us to propagate the candidate set of *top-k* aggressors in an organized manner. Furthermore, we proposed the dominance of noise envelopes which imposes a partial ordering on the aggressors and enables us to efficiently prune the enumeration space. We implemented an implicit enumera-

tion algorithm for identifying the set of *top-k* aggressors. Experimental results show that the proposed algorithm achieves a significant speed-up without compromising accuracy over the brute-force enumeration approach.

CHAPTER V

Modeling Crosstalk in Statistical Static Timing Analysis

Imprecise control of lithography equipment and channel doping can lead to a significant variability of the device dimensions and threshold voltages. In nanometer process technology , variability in manufacturing process has not scaled commensurate with the device dimensions. Consequently, variability of circuit performance has been rapidly increasing as we continue to shrink the device dimensions. To account for variability in the timing verification of the circuit, we can perform traditional static timing analysis (STA) at multiple process, voltage and temperature (PVT) corners. However, with an increase in variability, the number of corners needed to accurately model the circuit performance has grown rapidly. Therefore, statistical static timing analysis ($SSTA$), which models gate delays and circuit performance as random variables with a probability distribution function (PDF), has emerged as an efficient alternative to corner-based STA. Most of the techniques proposed in SSTA can be classified as either path based or block based. Path-based SSTA algorithms [5, 53] compute the delay distribution of the critical paths in the circuit, and accurately preserve the correlation information between paths. However, path-based approaches suffer from an explosion in the total number of paths that have to be enumerated. On the other hand, block-based SSTA [4, 19, 72] requires only a single

PERT-like traversal of the circuit graph and is more efficient than path-based SSTA.

Scaling of device dimensions has also led to a considerable reduction in gate delays. However, due to less aggressive interconnect scaling, wire delays have not reduced in proportion to gate delays, and wire delays, especially the global-interconnect delays, now contribute significantly to the total circuit delay. Due to capacitive coupling between wires, wire delay depends on the switching activity of neighboring wires. As the spacing between wires continues to shrink, the magnitude of the coupling capacitance increases and dominates the wire ground capacitance. Therefore, the magnitude of noise that is coupled on a victim net due to switching transitions of aggressor nets has become significant. If the aggressor-victim pair switch in the same direction, then the coupling noise can speed up the victim transition and reduce the victim-stage delay. On the other hand, if the aggressor-victim pair switch in mutually opposite directions, then the coupling noise can slow down the victim transition and increase the victim-stage delay. This change in the victim-stage delay due to coupling noise is referred to as delay noise and it contributes to a significant portion of the total circuit delay. Therefore, accurate modeling of delay noise is necessary for accurate timing analysis of VLSI circuits.

It has been observed that delay noise strongly depends on the aggressor-victim input skew or the difference between arrival times at the inputs of aggressor-victim drivers. Process variations translate into delay variations, and the delay variability of upstream gates translates into uncertainty in the arrival times at the input of aggressor-victim drivers. Therefore, due to variability in aggressor-victim input skew, delay noise can no longer be treated as a deterministic quantity. Sources such as the aggressor-victim interconnect variation also contribute to variability of the delay noise. However, a majority of the timing-analysis techniques used today model the

delay noise on the victim as a deterministic quantity.

In [37], the overlap between the aggressor-victim timing windows, was used to identify whether the aggressor can couple noise on the victim. In block-based SSTA, however, the end points of statistical timing windows are random variables obtained by performing recursive *max* and *min* atomic operations in a topological order. In [48], the authors extend the above idea to SSTA by expanding the *nominal* timing window by 3σ on both sides, where σ is the standard deviation of early and late arrival times. Overlap between the *expanded* timing windows of the aggressor-victim pair is used to identify whether noise is coupled on the victim net. Since, the worst-case delay noise is applied whenever the expanded windows overlap, the above technique leads to a pessimistic estimation of delay noise.

The mutual dependence of delay noise and timing windows leads to a *chicken-and-egg* problem. However, in [68] the authors propose an iterative approach for crosstalk-aware SSTA as a fixpoint solution on a lattice and theoretically proved its convergence. In [71], the coupling capacitance is modeled as a random variable which depends on the skew between aggressor-victim arrival times. In [45], the authors provide a closed-form expression for computing the PDF of delay noise given the aggressor-victim input arrival-time distributions. However, delay noise was assumed to be independent of the input arrival-time distributions and no correlation information of the delay noise was preserved. The lack of correlation information makes it difficult to integrate the delay-noise distribution accurately into SSTA tools.

A canonical first-order model was used in [19, 72] to capture the first-order sensitivities of delay to the normally distributed sources of variation and preserve the correlation information among all timing quantities. In this work, we show how the first-order SSTA framework can be extended to accurately capture the variation in

delay noise. We model the statistical delay-noise distribution as a random variable and express it in the canonical form by computing its first-order sensitivities to the variation sources. Given a Delay Change Curve (DCC) that captures the dependence of delay noise on the aggressor-victim input skew and an input-skew distribution in canonical form, we obtain closed-form expressions of the resulting delay-noise distribution. To compute the noisy victim-output arrival time, we must add the delay noise to the noiseless victim-output arrival time. We express delay noise in the canonical form by matching the first two moments and computing the correlation information. Since delay noise and victim-output arrival time are both expressed in canonical form, we can use the statistical *sum* operation to compute the noisy victim-output arrival time.

In this chapter, we propose the use of a statistical skew window whose end points are obtained by subtracting the end points of the aggressor-input timing window from the late victim-input arrival time. Using the skew window and the DCC, we analytically obtain the delay-noise distribution in canonical form. We propose to fragment the skew window into smaller segments to further reduce pessimism in our analysis. Using the fragmented skew window and the DCC, we then obtain the distribution of delay noise. The proposed technique matches well with Monte-Carlo simulations, and we observe a significant reduction in pessimism compared to prior approaches which do not model delay noise as a statistical quantity. The rest of the chapter is organized as follows: In Section 5.1, we analyze the problem of computing the delay-noise distribution in the presence of process variations. In Section 5.2, we present an analytical technique to compute the delay-noise distribution in canonical form, given a single aggressor-victim input-skew distribution and a DCC. In Section 5.3, we extend the analytical technique such that statistical-delay noise computation

can be performed within the SSTA framework with statistical timing windows. In Section 5.4, we present experimental results, and in Section 5.5 we conclude this chapter.

5.1 Problem Description

In this section, we examine the problem of modeling the delay-noise distribution in the presence of process variations. The amount of delay noise coupled to a victim by an aggressor depends on several factors such as the aggressor-victim slew rates, the driver strengths, the coupling and the ground capacitances, and the input skews. Also, an aggressor can couple noise only when its transition is temporally close to the victim transition. Therefore, the magnitude of delay noise strongly depends on the aggressor-victim input skew. The HSPICE simulation plot in Figure 5.1 shows the delay noise as a function of the input skew and is referred to as the Delay Change Curve (DCC). The DCC can be derived by either using SPICE based methods [64] or using analytical methods [7] where the coupling-noise pulse on the victim is approximated by a two piece model, and the DCC is obtained analytically by curve-fitting techniques. Process variation leads to uncertainty in signal arrival times at the aggressor-victim inputs. Therefore, delay noise, which is a function of the input skew, is no longer deterministic. However, a majority of statistical timing analysis techniques model the worst-case delay noise as a deterministic quantity and can often lead to pessimistic results.

The goal of this work is to model delay noise in current SSTA framework where delays are expressed in a canonical form,

$$d = d_0 + \sum_{i=1}^{N} s_i \cdot p_i + s_{N+1} \cdot R, \tag{5.1}$$

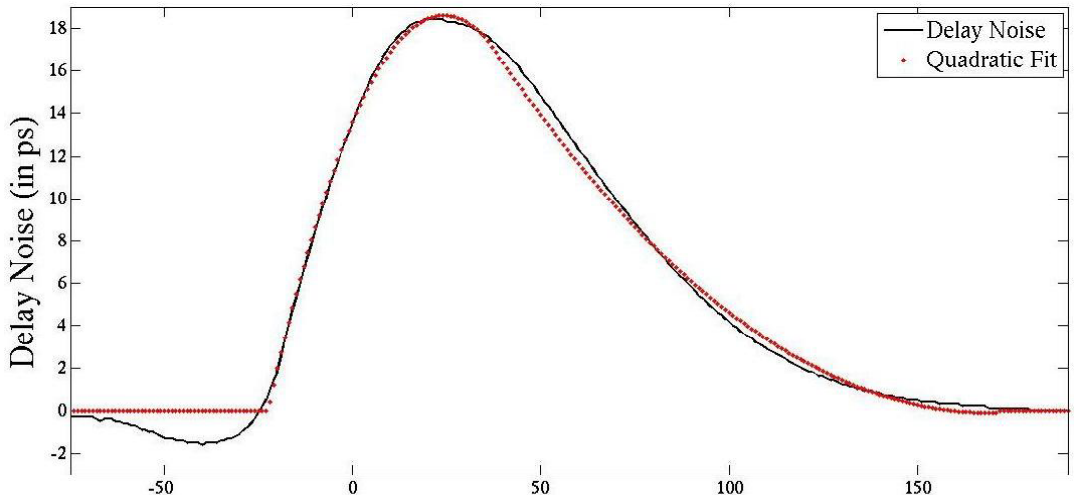


Figure 5.1: Delay change curve captures the dependence of delay noise on input skew.

where d_0 is the nominal delay, s_i is the sensitivity of delay to the process parameter p_i which is a standard normal random variable, R is the random component of the delay-noise distribution and is also a standard normal random variable. Principal Component Analysis (PCA) can be used to transform the set of process parameters into a set of mutually independent normal random variables. The early and late arrival-time distributions are propagated by performing statistical *min* and *max* operations recursively in a topological order [19, 72].

In Chapter 2, it was shown that the worst-case delay noise occurs when the victim-input transition occurs at the latest point in its timing window. Therefore, for computing the worst-case delay noise, we are only interested in the distribution of late victim-input arrival time. Given the statistical timing window at the input of the aggressor, we subtract the early and late aggressor-input arrival-time distributions from the late victim-input arrival-time distribution to obtain the skew window (as shown in Figure 5.2). In this work, using this statistical skew window and the

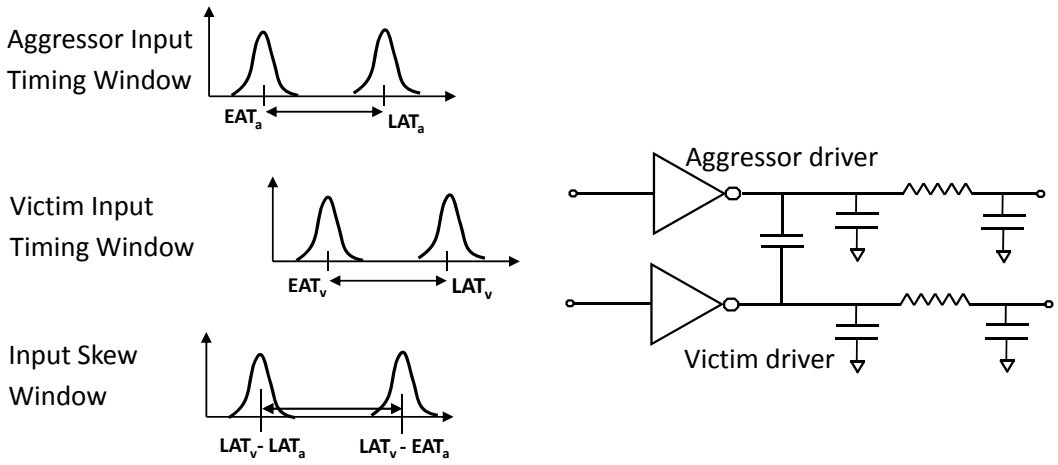


Figure 5.2: Skew window obtained by subtracting the early and late aggressor-input arrival times from the late victim-input arrival time.

DCC, we derive closed-form expressions for the mean and variance of the delay-noise distribution.

Note that the use of a single skew window can lead to pessimistic bounds on the delay-noise distribution. Therefore, we propose to divide the skew window into smaller segments to further reduce the amount of pessimism in our analysis. Since delay noise strongly depends on the input skew, in this work, we model the dominant source of variation that is variability in the input skew. Other sources such as variation in the aggressor-victim interconnects also contributes to variability of the delay noise. However, their contribution to delay-noise variability can be very small. For instance, it has been reported in [6], that interconnect variation causes only a 10% ($3\sigma/\mu$) variability in the peak of the coupling-noise pulse. Also, in [45] the authors show that variability in delay noise due to other sources of variation can be assumed to be independent of the input-skew distribution, without loss of accuracy. Therefore, in this work we focus on the dominant source of variation in delay noise that is the variation in the input-skew distribution. Also, the *chicken-and-egg* problem occurring due to the mutual dependence of delay noise and timing windows can be solved using

iterations [68]. Hence, in this work we focus on accurately modeling the delay-noise distribution on the victim within a single iteration of the delay-noise computation.

5.2 Statistical-Delay Noise

In this section, we analytically compute the delay-noise distribution in canonical form, given a *single* input-skew distribution in canonical form and a quadratic model of the DCC. We first show that the relative ratios' of the sensitivities of delay-noise distribution is the same as that of the input-skew distribution. We then obtain closed form expressions for computing mean and standard deviation of delay-noise distribution. The results obtained in this section will be used later in Section 5.3 to compute the worst-case delay noise in SSTA framework where we have statistical skew windows.

5.2.1 Correlations in Delay Noise

As delay noise depends on input skew, the delay-noise distribution is correlated with the input-skew distribution. In this subsection, given a quadratic DCC model, we show that the correlations in the input-skew distribution are preserved in the delay-noise distribution. This fact allows us to represent the delay-noise distribution in a canonical form. Delay-noise distribution in canonical form preserves the necessary correlation information and can easily be integrated in traditional block-based SSTA methods.

Theorem 5.1. *Given a quadratic DCC and an input-skew distribution in canonical form, the relative ratios' of the sensitivities of delay-noise distribution to process parameters is the same as that of the input-skew distribution.*

Proof. Without loss of generality, we assume an input-skew distribution s,

$$s = s_0 + s_1 \cdot x_1 + s_2 \cdot x_2, \tag{5.2}$$

having a mean s_0 and sensitivities s_1 and s_2 with respect to x_1 and x_2. Since x_1 and x_2 are *independent* and *standard* normal random variables, they have unit variances and zero cross-correlation.

The covariance of the input-skew distribution s with process parameter x_1 can be calculated as follows,

$$\begin{aligned} Cov(s, x_1) &= E[s - s_0, x_1] \\ &= E[s_1 \cdot x_1^2 + s_2 \cdot x_1 \cdot x_2] \\ &= s_1. \end{aligned} \tag{5.3}$$

where $E[\cdot]$ is the expectation operator. The delay noise obtained from the DCC has a quadratic dependence on the input skew s,

$$d = a \cdot s^2 + b \cdot s + c. \tag{5.4}$$

It is well-known that the odd moments of a standard normal random variable are zeros and the even moments evaluate to one. The mean of the delay-noise distribution (d) can be calculated as,

$$d_0 = E[d] = a \cdot (s_0^2 + s_1^2 + s_2^2) + b \cdot s_0 + c. \tag{5.5}$$

The covariance with parameter x_1 can be calculated as,

$$\begin{aligned} Cov(d, x_1) &= E[d - d_0, x_1] \\ &= E[a \cdot s_1^2 \cdot x_1^3 + a \cdot s_2^2 \cdot x_2^2 \cdot x_1 + 2a \cdot s_1 \cdot s_2 \cdot x_1^2 \cdot x_2 \\ &\quad + (2a + b) \cdot (s_1 \cdot x_1^2 + s_2 \cdot x_2 \cdot x_1)]. \end{aligned} \tag{5.6}$$

Since x_1 and x_2 are independent, the expectation of cross-product terms containing $x_1 \cdot x_2$ is zero. Also, the odd moments of a standard normal random variable are zeros and the even moments evaluate to one. Therefore, the cross-correlation terms all evaluate to zeros,

$$\begin{aligned} E[x_1^2 \cdot x_2] = E[x_1^2] \cdot E[x_2] = 1 \times 0 = 0, \\ E[x_1 \cdot x_2] = E[x_1] \cdot E[x_2] = 0 \times 0 = 0, \\ E[x_1 \cdot x_2^2] = E[x_1] \cdot E[x_2^2] = 0 \times 1 = 0, \\ E[x_1^3] = 0, \\ E[x_1^2] = 1. \end{aligned} \tag{5.7}$$

Using the linearity of expectation operator and the above result, we can simplify the expression of the covariance terms of Equation 5.6,

$$Cov(d, x_1) = (2a + b) \cdot s_1. \tag{5.8}$$

Note that the covariance of the delay noise obtained in Equation 5.8 is the same as the covariance of input-skew distribution obtained in Equation 5.3 multiplied by a constant factor (*i.e.*, $2a + b$). Performing a similar analysis with process parameter x_2, we obtain the following,

$$Cov(d, x_2) = (2a + b) \cdot s_2. \tag{5.9}$$

Since the covariance of delay noise with respect to process parameters are a scaled version of the covariance of the input-skew distribution, from Equations 5.3, 5.8 and 5.9,

$$\frac{Cov(d, x_2)}{Cov(d, x_1)} = \frac{Cov(s, x_2)}{Cov(s, x_1)} = \frac{s_2}{s_1}. \tag{5.10}$$

Since the ratios between covariance terms of the canonical delay-noise distribution and the input-skew distribution remains constant, the correlation information in the input skew is preserved in delay noise. Note that this result is independent of the number of process parameters considered in the input-skew distribution as every covariance is scaled by the same factor. □

Given an input-skew distribution and a quadratic DCC, using Theorem 5.1 we can obtain the correlations of the delay-noise distribution. To express the delay-noise distribution in canonical form, we only need to compute the mean and the standard deviation of the delay-noise distribution.

5.2.2 Canonical Delay-Noise Distribution

In this subsection, given a quadratic model of the DCC and the input-skew distribution in canonical form, we analytically compute mean and standard deviation of the delay-noise distribution. Suppose that the input-skew distribution is given by Equation 5.2. Since the process parameters are normal random variables, the input-skew distribution f_s is therefore normally distributed with mean μ and standard deviation σ given by,

$$f_s(s) = N(\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \cdot exp(-\frac{(s-\mu)^2}{2\sigma^2}). \tag{5.11}$$

Suppose that the piece-wise quadratic DCC has the following functional form,

$$d_{cc} = \begin{cases} 0, & s < z_0, \\ a_1 s^2 + b_1 s + c_1, & z_0 \leq s \leq z_1, \\ a_2 s^2 + b_2 s + c_2, & z_1 \leq s \leq z_2, \\ 0, & s > z_2. \end{cases} \tag{5.12}$$

Since the delay noise is a function of only the input skew and the functional dependence is captured by the DCC, we can appeal to the basic theory of probability and statistics [54] to obtain the PDF of delay noise as a function of the input-skew distribution,

$$f_d(s) = \frac{f_s(x_1)}{|2a_1 x_1 + b_1|} + \frac{f_s(x_2)}{|2a_2 x_2 + b_2|}. \tag{5.13}$$

where x_1 and x_2 are roots of the two quadratic pieces of the DCC, respectively,

$$\begin{aligned} x_1 &= \frac{1}{2a_1} \cdot (-b_1 + \sqrt{b_1^2 - 4a_1(c_1 - s)}), \\ x_2 &= \frac{1}{2a_2} \cdot (-b_2 + \sqrt{b_2^2 - 4a_2(c_2 - s)}). \end{aligned} \tag{5.14}$$

Note, that the delay-noise distribution obtained in Equation 5.13 is not necessarily gaussian. However, as observed in [45], the delay-noise distribution behaves like a normal distribution when the variance σ^2 of the input-skew distribution is small and the mean μ falls on the linear part of the DCC. Using the PDF of delay distribution from Equation 5.13, it is possible to compute the first and the second moments of the delay-noise distribution in closed form (refer to Appendix B). The delay-noise distribution in the canonical form can finally be constructed by matching

the first two moments of delay-noise distribution. Correlations of the delay-noise distribution are assigned by using the sensitivities of the input-skew distribution to process parameters.

5.3 Delay Noise in SSTA

In the previous section, we obtained closed-form expressions for the mean and variance of the delay-noise distribution given a single input-skew distribution. It was also observed that the correlations in the input skew are preserved in the delay-noise distribution. However, in block-based $SSTA$ framework, we no longer have a single skew distribution at the input of the victim driver that can be used to compute the delay-noise distributions. Instead, we have statistical timing windows at every node where the early and late arrival times are canonical distributions obtained by performing statistical *min* and *max* operations, respectively. In this section, we propose the use of a statistical skew window for computing the delay-noise distribution. To further reduce pessimism in our analysis, instead of using a single skew window, we propose the use of multiple smaller statistical skew windows. We look at the computation of a skew window for an aggressor-victim pair, and explain how it can be used to compute the corresponding delay-noise distribution.

5.3.1 Delay-Noise Distribution using Skew Window

In chapter 2, it was shown that regardless of the aggressor transition, the worst-case delay noise occurs when the victim-input transition occurs at the latest point in its timing window. In other words, adding the worst-case delay noise to the latest victim-input arrival time will result in the maximum slow-down (increase in the victim-output timing window). Therefore, for computing the worst-case delay noise, we are only interested in the distribution of late victim-input arrival time. Given the

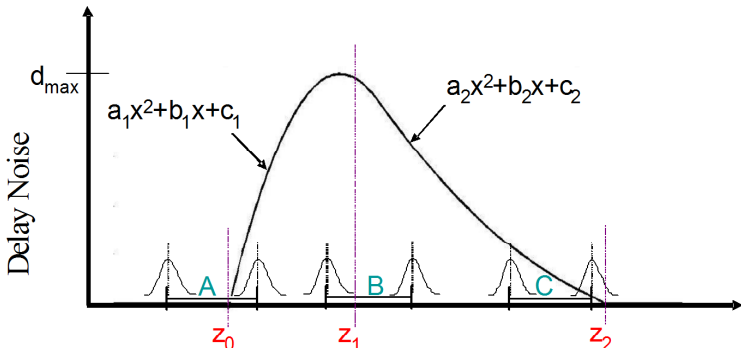


Figure 5.3: Delay change curve in piece-wise quadratic form.

statistical timing window at the input of the aggressor, we subtract the early and late aggressor-input arrival-time distributions from the late victim-input arrival-time distribution to obtain the skew window. The arrival-time distributions of end points of the skew window are referred to as early and late skew distributions.

The arrival times are normal random variables in canonical form. Note that the mean of the difference of the two normal distributions is given by the differences in their individual means. Therefore, the skew window that was obtained by using the early and late arrival times of the aggressor bounds the mean of any feasible input-skew distribution. This observation is true because there exists no aggressor-input arrival-time distribution whose mean is greater than the mean of the late arrival-time distribution, or less than the mean of early arrival-time distribution. Since we always use the late victim arrival-time distribution to compute the skew window, we conclude that the mean values of all feasible skew distributions must lie within the skew window.

We look at how the skew window can be used to compute the delay-noise distribution. As shown in Figure 5.3, the skew window can align with the *DCC* in three different ways. Case *A* occurs when the mean of late-skew distribution is less than the worst-case skew value of the *DCC* (i.e., z_1). Case *B* occurs when the mean

of late-skew distribution is greater than z_1 and the mean of early-skew distribution is less than z_1. Lastly, Case C occurs when the mean of early-skew distribution is greater than z_1. Since any skew distribution which lies within the skew window is feasible, for Case B, the delay noise is modeled by its worst-case value (d_{max}). Note that in Case A, the DCC is an increasing function of skew. Hence, for any feasible skew distribution with a variable mean and fixed variance, the corresponding mean of the delay-noise distribution will increase as we shift the mean of input-skew distribution to the right. Therefore, the mean of the delay-noise distribution will be maximized when the mean of the feasible skew distribution coincides with that of the late-skew distribution. Using the above fact, we propose to use the analytical results from Section 5.1 and the late-skew distribution to compute the delay-noise distribution in canonical form. Similarly, for Case C, we use the early-skew distribution to obtain the delay-noise distribution. Thus, given the statistical timing windows from block-based $SSTA$, we can analytically compute the delay-noise distribution. Also, since the delay-noise distribution is computed in the canonical form, we can use statistical *sum* operation and add it to the late victim-output arrival-time distribution to obtain the noisy output arrival-time distribution.

5.3.2 Multiple Skew Windows

Delay-noise computation by using the skew window assumes a worst-case delay noise for the Case B. This worst-case assumption could prove to be pessimistic, especially when there are only a few paths terminating at the input of the aggressor. For every path that terminates on the aggressor, we obtain a corresponding skew distribution by subtracting the path-delay distribution from the late victim-input arrival-time distribution (as shown in Figure 5.2). As pointed out earlier, the mean values of each of these skew distributions is bounded by the skew window. Suppose

we arrange the skew distributions in the order of increasing mean. If the number of paths terminating on the aggressor are small, then it is possible that there is a significant gap between the means of two consecutive paths. Under such circumstances, the probability of the occurrence of the worst-case delay noise can be reduced considerably.

Instead of using a single skew window, we propose using multiple skew windows [23] as a technique to reduce the amount of pessimism in the computation of delay-noise distribution. Suppose we fragment the single skew window, which starts at the mean of early-skew distribution and ends at the mean of the late-skew distribution, into five smaller skew windows. If we have a path whose mean delay falls within any of the five smaller skew windows, then we assign the path to that particular skew window. In other words, the mean of the path distribution is bounded by the smaller skew window. Therefore, the end points of the smaller skew window are characterized by the path delay distribution. These smaller skew window can be used exactly in the same manner as earlier (i.e., Case A, Case B and Case C). This approach of using multiple skew windows allows us to identify those cases where the worst-case skew is not feasible due to fewer number of paths terminating on the aggressor.

5.4 Experimental Results

In this section, we will show experimental results that verifies the accuracy and effectiveness of our proposed approach for modeling the delay-noise distribution. The accuracy of the analytical results for computing mean and standard deviation of the delay-noise distribution is first verified against those obtained from Monte-Carlo simulations in Figure 5.4. A normal input-skew distribution is assumed whose standard deviation is fixed at $10ps$ and whose mean is varied from $-50ps$ to $200ps$. The

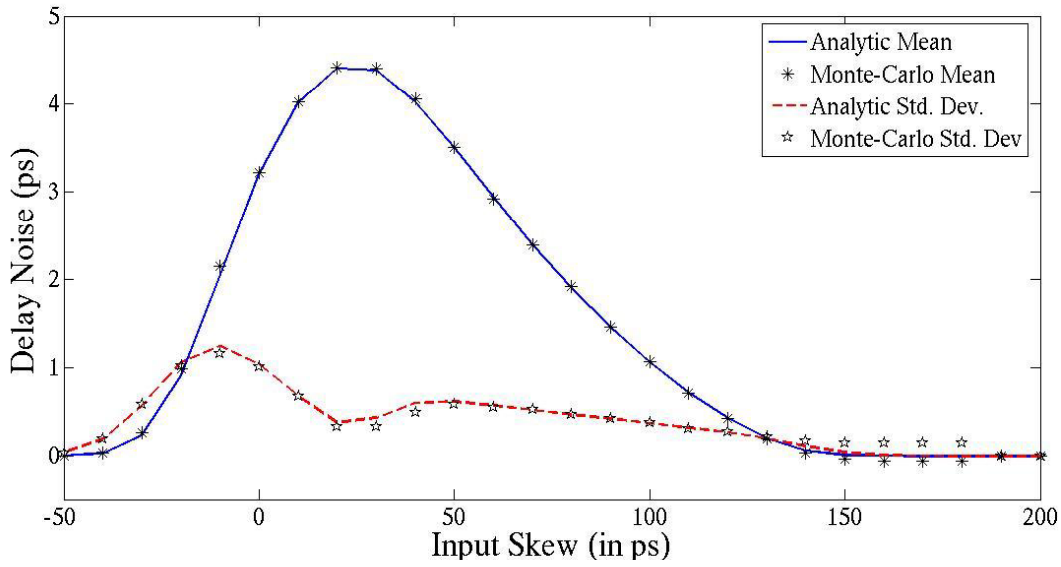


Figure 5.4: Comparison of analytical delay-noise distribution with that obtained from Monte-Carlo simulations.

mean and the standard deviation of the delay-noise distribution was computed using the *DCC* in Figure 5.1 and the analytical results from Section 5.2. The accuracy of the results are verified against Monte-Carlo simulations by using 10,000 different samples of the input-skew distribution. As expected, the mean and standard deviation values computed analytically matches very closely with that predicted from the Monte-Carlo simulations. Also, it can be seen that the mean delay-noise peaks when the input skew is aligned at the worst-case skew value of 30 ps (see Figure 5.1).

5.4.1 Results on Benchmark Circuits

A 130 nm standard-cell library was used for synthesis and technology mapping of LGSynth91 benchmark circuits [1]. The designs were placed-and-routed by using the Cadence Silicon Ensemble tool with a target row utilization of 80% and using up to three metal layers. The distributed RC interconnects were extracted by using Mentor Graphics Calibre tools. A prototype noise-analysis tool was implemented in C++, which used the extracted coupling-capacitances values to compute the delay noise in

Table 5.1: Pessimism reduction in the circuit delay with statistical noise analysis.

	Nominal circuit delay (in ps)		Circuit delay with worst-case delay noise (in ps)		Circuit delay with statistical delay noise (one skew window)(in ps)		Circuit delay with statistical delay noise (ten skew windows)(in ps)	
ckt	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.
i1	458.56	12.59	678.7	11.9	576.1	11.3	554.6	11.50
i2	595.14	20.1	891.2	18.8	743.75	19.8	718.8	19.97
i3	445.4	3.93	659.1	3.63	564.1	2.94	543.9	3.68
i4	683.2	14.32	979.5	15.5	852.5	14.52	818.9	14.64
i5	780.9	5.48	1234.9	7.54	979.1	6.95	932.0	6.92
i6	734.4	18.5	1347.4	12.1	968.31	7.42	916.4	6.64
i7	701.5	28.7	1344.5	28.4	964.1	14.1	911.1	14.7
i8	836.2	25.4	1253.3	22.8	1052.3	23.2	998.3	21.4
i9	1047.3	44.78	1677.5	44.9	1251.7	47.8	1182.5	47.96
i10	2424.4	65.15	3136.2	58.7	2640.2	61.15	2558.8	61.25

the circuit. For every aggressor-victim pair, the DCC were analytically generated on the fly. Process variations were modeled by assuming a $3\sigma/\mu$ variability of 30% in the gate lengths of the aggressor-victim drivers.

In Table 5.1, we show the circuit-delay distribution obtained by incorporating coupling noise in block-based SSTA. Columns 2 and 3 report the mean and standard deviation of the nominal circuit delay obtained when there is no coupling noise. Columns 4 and 5 report the worst-case circuit delay obtained by using the approach suggested in [48] and assuming worst-case delay noise whenever the statistical aggressor-victim timing windows overlap. Note that the circuit delay obtained by using the worst-case analysis can be very pessimistic, e.g., the mean circuit delay of the largest benchmark circuit $i10$ increases by about 29% compared to the mean of the nominal circuit-delay distribution with no coupling noise.

Columns 6 and 7 report the circuit-delay distribution obtained by modeling the statistical delay-noise distribution using the proposed approach with a single statisti-

cal skew window. The proposed approach results in a significant pessimism reduction compared to worst-case noise analysis. For instance, the percentage increase in mean of the circuit-delay distribution for circuit $i10$ is less than 9% of its nominal circuit delay. In the last two columns, we see that the use of ten smaller skew windows instead of a single skew window leads to a further reduction in the mean of the circuit-delay distribution. With the usage of smaller skew windows, the percentage increase in the mean of the circuit-delay distribution for circuit $i10$ is less than 6% of the nominal circuit delay. Hence, performing statistical noise analysis with a single skew window leads to a reduction in the mean circuit delay noise by 69.7% compared to worst-case noise analysis. Furthermore, we obtain an additional reduction of 11.43% in the mean circuit delay noise by the usage of ten smaller skew windows for every aggressor. In Figure 5.5, we show a bar chart of the percentage reductions in the circuit delay noise for all the benchmark circuits.

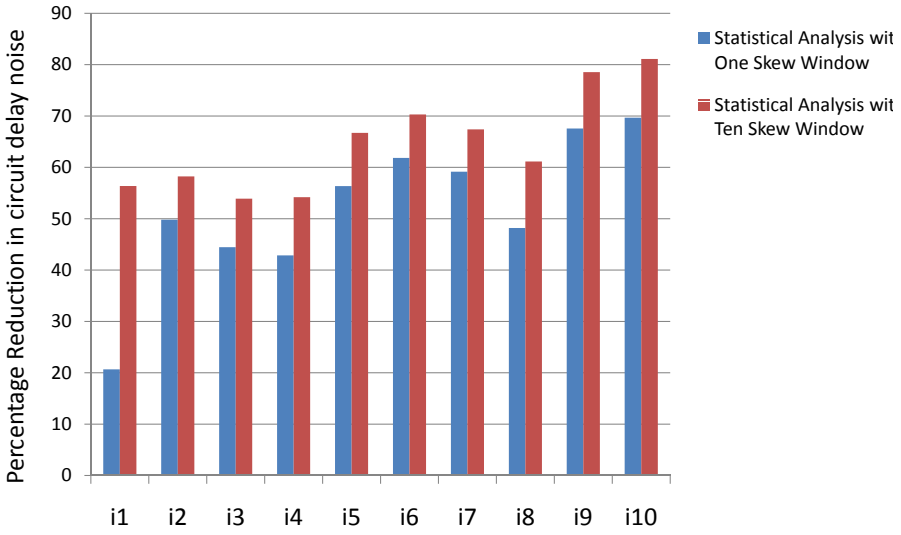


Figure 5.5: Percentage reduction in circuit delay noise with statistical noise analysis.

5.5 Summary

In this work, we model the variability in the delay noise which occurs due to variability in the aggressor-victim alignment. We analytically compute mean and standard deviation of the delay-noise distribution. We also proved that, for a quadratic DCC, the correlations in delay-noise distribution is the same as that of the input-skew distribution. Using the correlation information and matching the first two moments, we represent the delay-noise distribution in the canonical form and can be integrated into statistical timing analysis tools. The accuracy of the analytical results was verified against Monte-Carlo simulations. It was shown that statistical noise analysis leads to a significant pessimism reduction compared to worst-case noise analysis.

CHAPTER VI

Pessimism Reduction with Path-Based Statistical Noise Analysis

We start this chapter by describing in detail the problem of computing the path delay by accounting for the effects of crosstalk noise. Figure 6.1 illustrates an arbitrary signal path in a circuit comprised of cascaded logic gates in between the launch and capture flip-flops. The amount of time required to propagate a switching transition from the output of the launch flip-flop to the input of the capture flip-flop is referred to as the path delay. It can be noted that the path delay can be obtained by summing together the delays of all the logic gates and the interconnects present in the signal path. While performing path-based timing analysis, the path delays of all the *critical* paths in the circuit are verified to be lesser than the clock-cycle period, which is derived from the chip frequency. Consequently, a timing violation is flagged when the path delay is greater than the required clock-cycle period. If any timing violations are detected, then the circuit designer needs to redesign or fix these paths such that all the timing constraints are satisfied.

Now, every stage of a signal path could be coupled to crosstalk aggressors. For instance, in Figure 6.1, the victim net v is coupled to the three aggressors $a_1 - a_3$. Due to charge transfer through the coupling capacitances, the switching characteristics of the nets on the signal path are affected by the simultaneous switching of the

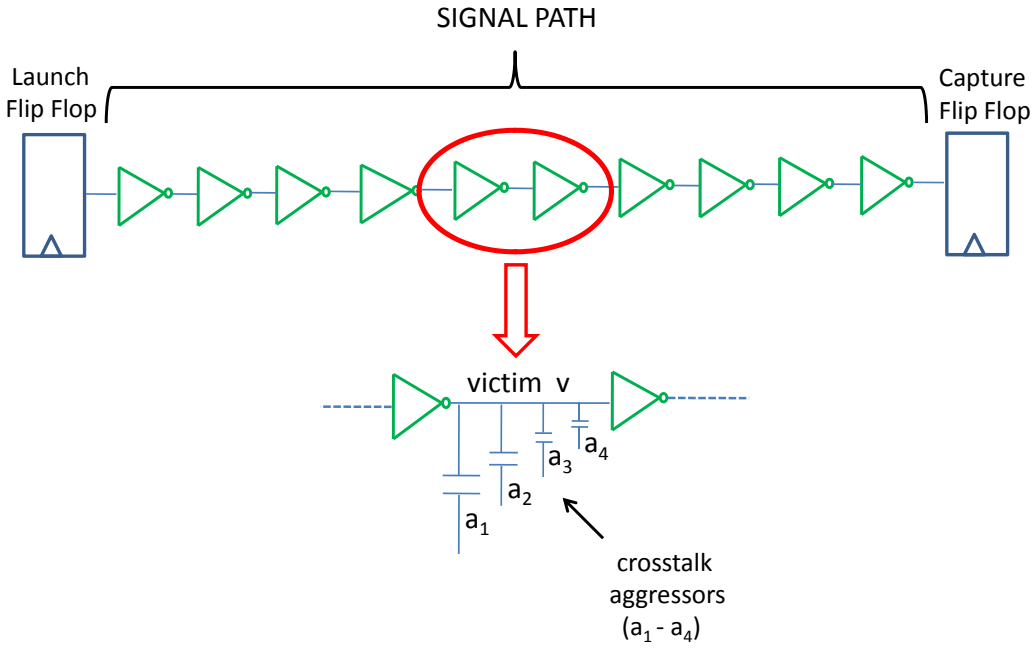


Figure 6.1: Example of a signal path with coupled aggressors.

crosstalk aggressors. The coupling noise injected by the aggressors can either increase or decrease the stage delay of the victim depending on the mutual aggressor-victim switching directions. If the aggressor-victim pair switch in the same direction, then the coupling noise can speed up the victim transition and reduce the victim-stage delay. On the other hand, if the aggressor-victim pair switch in mutually opposite directions, then the coupling noise can slow down the victim transition and increase the victim-stage delay. This change in the victim-stage delay due to capacitive coupling is referred to as *delay noise*. Similarly, the change in the path delay due to all the aggressors coupled to the signal path is referred to as the *path delay noise* and can be obtained by adding together the delay noise of all the victims present in the signal path. Typically, worst-case noise analysis reports the maximum possible path delay noise. Hence, during worst-case noise analysis, it is assumed that all the

aggressors coupled to the signal path are switching in the opposite direction of their corresponding victim nets.

However, using the worst-case delay noise contribution from all the crosstalk aggressors coupled to the signal path is often very conservative. In any given clock cycle, depending on the circuit topology and the stage of the input vectors, the crosstalk aggressors which are coupled to the signal path

- may not switch at all,
- may switch in the same direction as their corresponding victims,
- may switch without any temporal overlap with the victim transition.

Consequently, all the above aggressors do not result in a slow down of their respective victim transitions. Hence, assuming the worst-case delay noise contribution from all the aggressors coupled to a signal path can be pessimistic. Consider an example where four crosstalk aggressors are coupled to every stage of a signal path comprising of ten logic gates (as shown in Figure 6.1). Suppose that the probability of a given crosstalk aggressor switching in a clock cycle is 50%, and that the probability of it switching in the opposite direction of the corresponding victim is 50%. Assuming independence between the switching of aggressors, the probability of the occurrence of the worst-case path delay noise can be computed to be $(0.5 * 0.5)^{-4*10}$. One can note that the worst-case noise event, with all aggressors switching in opposite directions of their respective victims, has a very small probability of occurrence with an unreasonably large mean-time-to-failure of $3.84 * 10^7$ years on a one gigahertz machine.

The simple example above illustrates the fact that the worst-case path delay noise has a very low probability of occurrence, especially for critical paths having

several logic stages and multiple crosstalk aggressors coupled to each logic stage. Hence, assuming the worst-case path delay noise could be very conservative. We illustrate the pessimism associated with worst-case noise analysis by analyzing the path delay-noise distribution of a signal path that belongs to an industrial circuit and is comprised of 8 cascaded logic stages. Noise analysis was performed in the 65nm technology node using an industrial timing tool (PrimeTime SI). A total of 10,000 samples were used in performing the Monte-Carlo (MC) simulations in which the transitions of the aggressors coupled to the signal path were treated as random variables. In each sample, the switching probability of every aggressor was assumed to be 50%. Also, it was assumed that every switching aggressor has equal probability of having either a rising or a falling transition. In each MC sample, the delay noise contributed from all the aggressors switching in an opposite direction of the victim were added to obtain the victim delay noise. The path delay noise is finally obtained by adding up the delay-noise contributions from all the victim nets on the signal path. In Figure 6.2, we plot the path delay-noise distribution, and also plot the worst-case path delay noise. It can be seen that the worst-case path delay noise (84ps) is more pessimistic than the maximum delay noise value (68ps) of all the samples from the MC simulations. Hence, it can be seen that the worst-case assumption, which requires all the coupled-aggressors to switch in the opposite directions with respect to their respective victim transitions, can indeed lead to pessimistic results.

The simple experiment shown above motivates the need for a statistical noise-analysis framework that can compute a more accurate (and less pessimistic) estimate of the path delay noise. The potential benefits of adopting a statistical noise-analysis framework can be summarized as follows:

- A statistical noise-analysis (instead of the traditional worst-case) framework

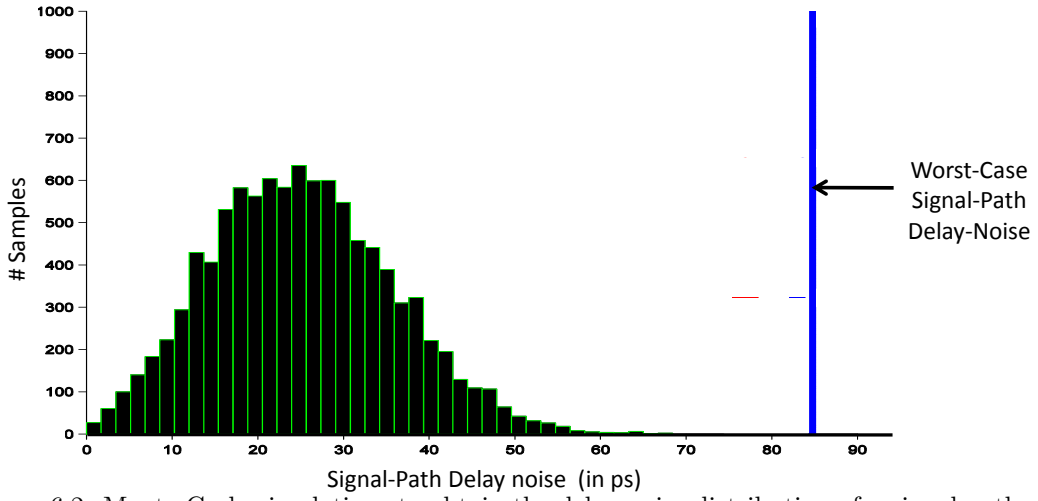


Figure 6.2: Monte-Carlo simulations to obtain the delay-noise distribution of a signal path.

could lead to a significant pessimism reduction in the delay noise of the circuit. Hence, it could potentially lead to a reduction in the number of crosstalk-noise violations reported while performing timing verification of the circuit. Therefore, the use of a statistical noise-analysis framework would lead to a faster design turnaround time as the circuit designers need to fix lesser number of crosstalk-noise violations.

- Worst-case noise analysis always leads to a pessimistic estimate of crosstalk effects. This is a source of apprehension for the circuit designers, since there is no systematic manner to characterize the amount of pessimism that is present in the results. Hence, circuit designers are concerned by the amount of pessimism that can potentially be present in the delay noise of the circuit and would prefer noise-analysis techniques which are more accurate and less pessimistic. The use of statistical noise-analysis approaches could lead to pessimism reduction in the delay noise of the circuit.

In this chapter, we propose a statistical noise-analysis framework to estimate the confidence bounds on the path delay-noise distribution. The noise coupled from each aggressor is treated as a random variable for every victim lying on the signal path. Using the superposition assumption, the cumulative coupling-noise pulse on the victim is the sum of all the individual coupling-noise pulses from all the aggressors. Therefore, the cumulative coupling-noise pulse is a random variable, and we leverage the Hoeffding's inequality [41] to obtain an upper bound on the peak of the cumulative coupling-noise pulse. Using the computed bounds on the peak of the cumulative coupling-noise pulse, we finally obtain bounds on the delay noise of every victim lying on the signal path.

The remainder of the chapter is organized as follows: In Section 6.1, we present an approach to compute the confidence bounds on the path delay-noise distribution. In Section 6.2, we show experimental results which confirm the effectiveness of our approach, and in Section 6.3 we summarize the chapter.

6.1 Proposed Approach

In the previous section, we saw that the worst-case noise analysis can lead to a significant amount of pessimism in the delay noise of the signal path. This motivates the use of statistical noise analysis, which seek to estimate the confidence bounds on the path delay-noise distribution. In this section, we describe in detail the statistical noise-analysis framework which uses the Hoeffding Inequality to compute confidence bounds on the path delay-noise distribution.

6.1.1 Path Delay-Noise Distribution

As opposed to worst-case noise analysis, the proposed approach seeks to estimate the confidence bound on the path delay-noise distribution. Hence, the starting point

is to first quantify the confidence point of the path delay-noise distribution. Since, one expects these chips to function reliably over its normal lifetime duration, the proposed statistical noise analysis must be conservative. Therefore, the desired confidence point of the path delay-noise distribution must be very high, and we seek to bound the tail of the delay-noise distribution. We use a basic measure of reliability, the mean-time-to-failure (MTTF) metric which is defined as the expected time until the first failure of the chip occurs. Note that in the context of noise analysis, a failure is defined to occur in a particular clock cycle when the path delay noise exceeds the value of the computed bound on the path delay-noise distribution.

Given a chip MTTF (e.g., five years), the expected path failure probability (P_f^{path}) in a single clock cycle can easily be obtained as follows,

$$P_f^{path} = \frac{1}{Chip\ Frequency * Chip\ MTTF(in\ years) * 365 * 24 * 60 * 60}. \tag{6.1}$$

Suppose D_{noise}^{path} be a random variable denoting the path delay-noise distribution (an example histogram is demonstrated in Figure 6.2). Therefore, D_{noise}^{path} can be written down as the sum of the delay noise (D_{noise}^{stage}) on all the victim nets lying on the signal path,

$$D_{noise}^{path} = \sum_{k=1}^{k=N} D_{noise}^{stage,k}, \tag{6.2}$$

where N corresponds to the total number of victim nets present in the signal path. We know that the exact value of D_{noise}^{path} can change in every clock cycle, depending on the switching activity of the aggressors coupled to the signal path. The objective of statistical noise analysis is to estimate a bound (D_{bound}^{path}) on the path delay-noise distribution such that the following inequality holds,

$$Probability\ (D_{noise}^{path} > D_{bound}^{path}) \leq P_f^{path}. \tag{6.3}$$

In other words, we seek to obtain a bound D^{path}_{bound}, such that the probability of the actual delay-noise distribution D^{path}_{noise} exceeding D^{path}_{bound} is less than the path failure probability P^{path}_{f}. Typically, the path failure probability (as defined in Equation 6.1) is a very small quantity. In the case of a chip running at a frequency of 2.5 gigahertz, a MTTF of five years corresponds to a P^{path}_{f} of $2.53 * 10^{-18}$.

However, it must be noted that the path delay-noise distribution (D^{path}_{noise}) is a linear sum of the delay-noise distributions D^{stage}_{noise} of all victim nets lying on the signal path (as shown in Equation 6.3). Assuming independence between the noise coupled on different victim nets, we can divide the problem of computing a bound (D^{path}_{bound}) on the path delay-noise distribution into several smaller problems of computing bounds (D^{stage}_{bound}) on each victim delay-noise distribution. In other words, we seek to estimate the bounds on the delay noise (D^{stage}_{bound}) for all victim nets lying on the signal path such that they satisfy the following,

$$Probability\ (D^{stage,k}_{noise} > D^{stage,k}_{bound}) \leq P^{stage}_{f,k}, \quad \forall k \leq N. \tag{6.4}$$

where N corresponds to the total number of victims present in the signal path, and P^{stage}_{f} corresponds to the failure probability of every victim stage. If we assume independence between the coupling noises on the different victim nets of the signal path, then the victim-stage failure probability (P^{stage}_{f}) is related to the path failure probability (P^{path}_{f}) by the following,

$$\prod_{k=1}^{k=N} P^{stage}_{f,k} = P^{path}_{f}. \tag{6.5}$$

The statistical bound on the path delay-noise distribution (D^{path}_{bound}) is finally obtained by summing up the bounds on the delay noise (D^{stage}_{bound}) for every victim net

lying on the signal path,

$$D_{bound}^{path} = \sum_{k=1}^{k=N} D_{bound}^{stage,k}. \tag{6.6}$$

Hence, using the independence assumption, we were able to efficiently subdivide the problem of performing statistical analysis for the signal path into multiple smaller problem of performing statistical analysis for the victim stage. In the following subsection, we use Hoeffding's inequality to obtain the bounds on the victim delay-noise distribution.

6.1.2 Statistical Noise Analysis of the Victim Stage

In this subsection, we propose an approach to compute the bounds on the delay-noise distribution of a single victim net such that the probability of the delay-noise distribution exceeding the computed bound is less than P_f^{stage}. Let the victim net be coupled to K aggressors, and n_i be the coupling-noise pulse which occurs due to the switching transition of the i^{th} aggressor.

Using the principle of linear superposition, the cumulative coupling-noise pulse (C_n) on the victim is obtained by summing up individual coupling-noise pulses from all the switching aggressors,

$$C_n = \sum_{i=1}^{i=K} n_i. \tag{6.7}$$

In worst-case noise analysis, which results in the maximum D_{noise}^{stage}, all aggressors are assumed to switch in the opposite direction of the victim transition. Also, no restrictions are placed on the relative alignment of the coupling-noise pulses, and all the coupling-noise pulses (n_i) are positioned such that their peaks align at the same

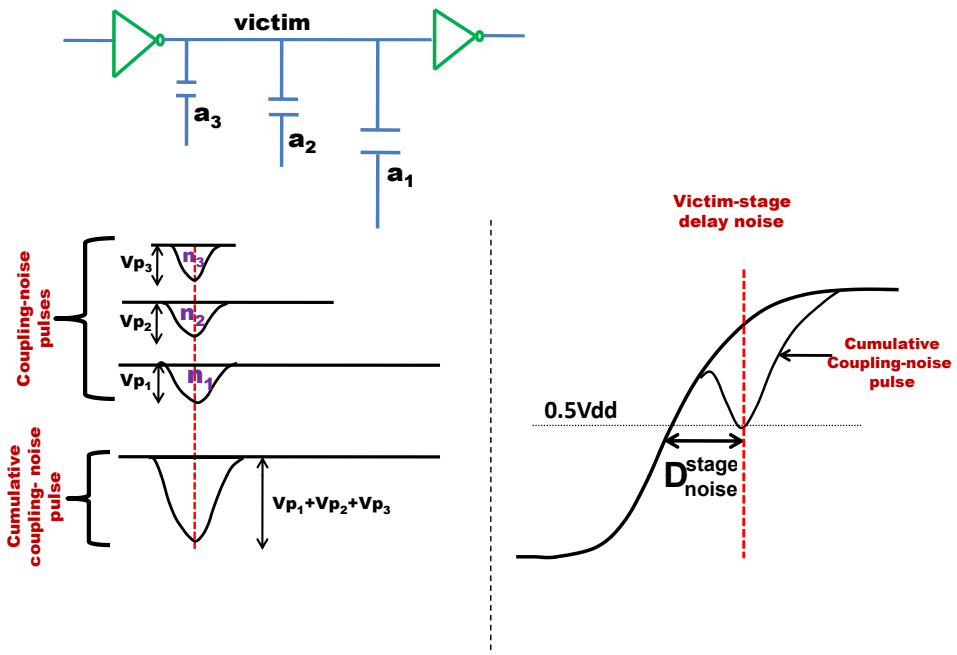


Figure 6.3: In worst-case noise analysis, delay noise is obtained by the linear superposition of the largest cumulative coupling-noise pulse, obtained with no restrictions on the alignment of aggressors, with the noiseless-victim transition.

time (as shown in Figure 6.3). Under this assumption, the coupling-noise pulses from all the aggressors add up together resulting in a cumulative coupling-noise pulse that has the largest noise-peak. Using the principle of superposition, the noise-peak of C_n can then be used with the noiseless-victim transition to obtain the worst-case delay noise on the victim (D_{noise}^{stage}) due to the switching of all the aggressors. It should be noted that, under the superposition assumption, the shape of C_n does not effect D_{noise}^{stage}. Instead, D_{noise}^{stage} only depends on the noise-peak of C_n.

In every clock cycle, the aggressor transitions are dependent on the input vectors and the circuit topology. Hence, it is improbable for all the aggressors to simultaneously switch in a direction opposite to the victim transition in the same clock cycle. Also, the size of the coupling-noise pulse changes in every clock cycle, since it is a function of the aggressor-victim slew rates which depend on the input vectors applied to the circuit. Furthermore, it is improbable for all the coupling-noise pulses

to occur such that their noise-peaks align perfectly. Hence, the coupling-pulse is a random variable whose noise-peak value is bounded by the worst-case value (Vp),

$$-Vp_i \leq n_i \leq Vp_i. \tag{6.8}$$

Therefore, in every clock cycle, the cumulative coupling-noise pulse (C_n) is a random variable whose noise-peak value is bounded by the worst-case value, and its exact value depends on the alignment of all the aggressor transitions. Therefore, combining Equations 6.7-6.8, we obtain the following bounds on C_n,

$$\sum_{i=1}^{i=K} -Vp_i \leq C_n \leq \sum_{i=1}^{i=K} Vp_i. \tag{6.9}$$

We seek to estimate a bound such that the following inequality holds,

$$Probability\ (C_n > C_n^{bound}) \leq P_f^{stage}. \tag{6.10}$$

Finally, using the computed bound (C_n^{bound}), we can obtain the bound on the victim delay noise (D_{bound}^{stage}), by superimposing with the noiseless-victim transition, such that the inequality of Equation 6.4 is satisfied.

In order to solve for C_n^{bound}, we use the Hoeffding's inequality [41] which is a result in probability theory that provides an upper bound on the probability of the sum of independent and bounded random variables to deviate from their expected values. The probability that the noise-peak value of C_n exceeds a certain value is bounded by the following inequality,

$$Probability\ (C_n > E[C_n] + S) \leq exp\left(-\frac{S^2}{2 \times \sum_{i=1}^{i=K} {Vp_i}^2}\right), \tag{6.11}$$

where S is any constant, and $E[C_n]$ is the mean noise-peak of C_n. The right hand side of above equation can be further simplified by choosing the parameter S to be

the following

$$S = k \times \sqrt[2]{\sum_{i=1}^{i=K} {Vp_i}^2}. \tag{6.12}$$

Using Equations 6.12-6.11, we obtain

$$Probability\ \left(C_n > E[C_n] + k \times \sqrt[2]{\sum_{i=1}^{i=K} {Vp_i}^2} \right) \leq exp(-\frac{k^2}{2}), \tag{6.13}$$

where the constant k can be expressed as a function of the victim-stage probability-failure (P_f^{stage})

$$k = \sqrt{-2 \times \log\left(P_f^{stage}\right)}. \tag{6.14}$$

Finally using the above expression of k in Equation 6.13, we obtain

$$Probability\ \left(C_n > E[C_n] + k \times \sqrt[2]{\sum_{i=1}^{i=K} {Vp_i}^2} \right) \leq P_f^{stage}. \tag{6.15}$$

Therefore, comparing the above expression with Equation 6.10, we finally obtain the bound on the noise-peak of the cumulative coupling-noise pulse (C_n^{bound}),

$$C_n^{bound} = E[C_n] + \sqrt{-2 \times \log\left(P_f^{stage}\right)} \times \sqrt[2]{\left(\sum_{i=1}^{i=K} {Vp_i}^2 \right)}. \tag{6.16}$$

Not surprisingly, C_n^{bound} depends on the properties of the cumulative coupling-noise pulse such as the mean ($E[C_n]$) and the root-mean-square (RMS) of the noise-peaks $\left(\sqrt{\sum_{i=1}^{i=K} {Vp_i}^2} \right)$. It also depends on the victim-stage failure probability P_f^{stage}, since C_n^{bound} should intuitively increase for smaller values of P_f^{stage}. All the values on the right hand side of Equation 6.16 are known except for the mean $E[C_n]$. We

propose a simple heuristic to estimate $E[C_n]$ by using the following,

$$E[C_n] = \sum_{i=1}^{i=K} p_i^{switching} \times p_i^{rise\_fall} \times Vp_i, \tag{6.17}$$

where $p_i^{switching}$ is the probability that the i^{th} aggressor switches in any clock cycle, and $p_i^{rise\_fall}$ is the probability that the i^{th} aggressor switches in the opposite direction of the victim transition. One could use vector-based simulation approaches or analytical techniques [36] to obtain these transition probabilities for the aggressor nets. The C_n^{bound} could be used to obtain the bound on the victim delay noise D_{bound}^{stage} by superposing with the noiseless-victim transition. Finally, We obtain the bound on the path delay-noise distribution by summing up the D_{bound}^{stage} values of all the victim stages lying on the signal path.

6.1.3 Worst-Case Alignment of Top-k Aggressors

The statistical noise-analysis methodology described in the previous section makes the assumption that the switching activity of all the aggressors coupled to the signal path are independent of each other. However, in a given clock cycle, the switching activity of each net in the circuit depends on the state of the input vectors and the circuit topology. Hence, the aggressor transitions could have a correlation that is not modeled in statistical noise analysis. For instance, the aggressors coupled to a victim stage could belong to a bus network and may have similar switching characteristics. Hence, the independence assumption of the aggressor transitions could in some cases lead to optimistic results. Therefore, we need a systematic technique to introduce additional pessimism into the bounds computed on the path delay noise (D_{bound}^{path}) by statistical noise analysis.

Additional pessimism could be introduced by performing worst-case noise analysis

on a subset of the aggressors (say the *top-k* aggressors, where k is a user-defined parameter) and statistical noise analysis on the rest of the aggressors. For example, in the circuit shown in Figure 6.4, the worst-case noise analysis can be performed on the $top-2$ aggressors (a_{1-2}), and statistical noise analysis is performed on the rest of the aggressors (a_{3-5}). One could use the noise-peak (Vp) values as the metric to screen the *top-k* aggressors coupled to the victim stage. Hence, one could use the proposed noise-analysis methodology where the amount of pessimism reduction compared to worst-case noise analysis is guided by the user-specified values of k.

6.2 Experimental Results

In this section, we show experimental results to verify the accuracy and effectiveness of the proposed approach for computing the confidence bounds on the path delay-noise distribution. The bounds on the path delay-noise distribution were computed, assuming a MTTF of five years and an equal failure probability (P_f^{stage}), for

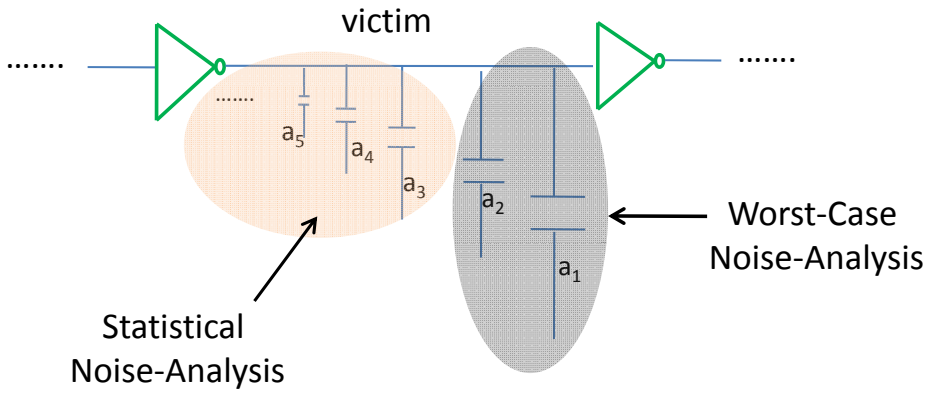


Figure 6.4: Perform worst-case noise analysis on the top-k aggressors and statistical noise analysis on the rest of the aggressors.

Table 6.1: Pessimism reduction in path delay noise by performing statistical noise analysis.

Circuit	# Gates	Avg # aggressors per victim	Delay noise as % of path delay	%Change in path delay noise (top-k=0)		%Change in path delay noise (top-k=1)		%Change in path delay noise (top-k=2)	
				MAX	AVG	MAX	AVG	MAX	AVG
ckt1	10,000	4.69	7.64	41.12	18.88	30.30	10.44	28.17	6.12
ckt2	0.75M	2.18	6.26	20.14	12.92	7.53	3.60	2.35	0.96
ckt3	0.28M	2.02	3.39	15.98	1.14	10.59	0.39	6.39	0.12
ckt4	1.1M	1.97	6.96	42.85	12.49	27.47	5.09	18.84	2.01
ckt5	0.9M	3.36	7.69	23.07	4.46	13.2	2.89	9.12	2.04

all the victims lying on the signal path. The switching probabilities defined in Equation 6.17 were assumed to be 0.5. All experiments were performed on industrial circuits in the 65nm technology node. As shown in the second column of Table 6.1, the circuit size varies from a few thousand (e.g., ckt1) to more than a million gates (e.g., ckt4). The circuits were physically implemented in the 65nm technology, and noise analysis was performed using the actual interconnect capacitances extracted from the designs. The industrial timing and noise analysis tool (Primetime-SI) was used to report the worst-case delay-noise values for the top-400 critical paths. We report in the third column, for each circuit, the average number of aggressors coupled to every victim net on the critical paths. The average number of aggressors per victim varies between $1.97 - 4.69$ for the experimental circuits shown in Table 6.1. The third column provides information about the average percentage contribution of the worst-case delay noise to the path delay. The worst-case delay noise contributes to more than 7% of the path delay for a few circuits (e.g., ckt1 and ckt5). The remaining columns report the pessimism reduction in the path delay noise that is obtained by performing statistical noise analysis.

It can be observed that we obtain a significant amount of pessimism reduction in path delay noise by performing statistical noise analysis. As expected, the pessimism reduction in delay noise is maximized when there are several aggressors coupled to

the critical path, i.e., maximum reduction can be observed for ckt1. In Columns 5 and 6, we report the maximum and the average reduction in the path delay noise obtained by performing statistical noise analysis over the 400 critical paths. Across all circuits, on an average, the proposed approach results in a 9.97% reduction in the path delay noise.

We also implement the mixed approach where worst-case noise analysis is performed on the *top-k* aggressors, and statistical noise analysis is performed on the remaining aggressors. The *top-k* aggressors of every victim stage are selected by choosing the ones that result in the largest coupling-noise peak (Vp) values from among all the coupled aggressors. In Columns 7 and 8, we report the maximum and the average reduction in the path delay noise when we perform worst-case noise analysis on the *top-1* aggressor and statistical noise analysis on the rest of the aggressors. As expected, this approach leads to a lesser average reduction in the path delay noise (4.48%). Finally, in Columns 9 and 10, we report the maximum and average reduction in the path delay noise when we perform worst-case noise analysis on the *top-2* aggressor and statistical noise analysis on the rest of the aggressors. In this case, we obtain the lowest reduction in path delay noise (2.25%). Finally, in Figure 6.5, we plot the histogram of the percentage reductions in the delay noise of the 400 critical paths for $ckt1$.

Traditionally, worst-case noise analysis has been used to compute the path delay noise, and it could result in several false timing violations because of the inherent pessimism. In contrast, statistical noise analysis provides an alternative framework

Table 6.2: Reduction in the total number of critical paths for ckt5 with statistical noise analysis.

No of Critical Paths (worst-case noise analysis)	No of Critical Paths (top-k=2)	No of Critical Paths (top-k=1)	No of Critical Paths (top-k=0)
933	922	870	780

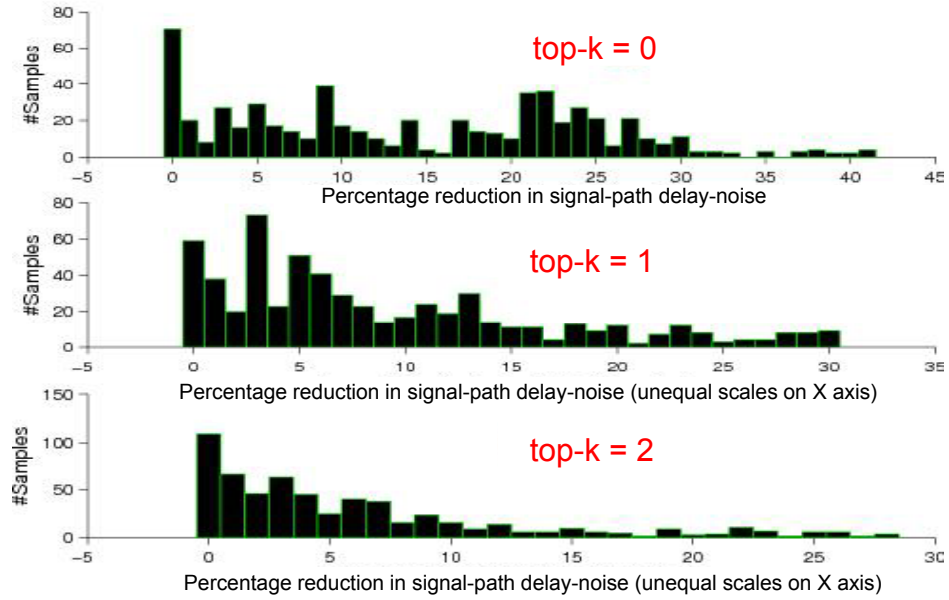


Figure 6.5: Histogram of the pessimism reduction in the delay noise of the top-400 critical paths for ckt1 (Note: unequal X-axis scales).

that can lead to a reduction in the amount of pessimism compared to worst-case noise analysis. In Table 6.2, we quantify the pessimism reduction by reporting the total number of critical paths for ckt5 that violate the timing constraints. Worst-case noise-analysis reports a total of 933 paths that could not meet timing requirements. However, worst-case analysis is pessimistic, and in comparison, the total number of violating paths reported by statistical noise analysis was reduced to 780.

6.3 Summary

In this chapter, we proposed a statistical framework for estimating the confidence bounds on the path delay-noise distribution. For every victim lying on the path, the noise coupled from each aggressor was treated as a random variable. Under the superposition assumption, the cumulative coupling-noise pulse on the victim was treated as a random variable. Using Hoeffding's inequality [41], an upper bound on the peak

of the cumulative coupling-noise pulse was obtained. Finally, using the computed bounds on the peak of the cumulative coupling-noise pulse, one could obtain the bounds on the delay noise of every victim lying on the path. Experimental results on industrial circuits show that the proposed statistical noise analysis approach leads to an average reduction of 9.97% in the path delay noise compared to worst-case noise analysis.

CHAPTER VII

Interconnect Corners Considering Crosstalk Noise

Imprecise control of photolithography equipment leads to a significant variability [16] in the geometry and the material properties of interconnects and devices, e.g., deviations in the implant dose can significantly affect the threshold voltages and the electrical parameters of a device [17, 76]. Similarly, the vertical and lateral dimensions of the interconnect wires are affected by variability in the metal deposition process and the dishing effects during metal etching. Furthermore, variability of the manufacturing process is expected to exacerbate as the feature sizes continue to shrink in future technology nodes. In the past, variability of the devices was the dominant source of variation in the circuit delay. However, with the ever-increasing contribution of the interconnect delay to the circuit delay, it has become necessary to accurately model the variability in the interconnects while performing timing verification of VLSI circuits.

The variations in the interconnect resistance and capacitances are correlated, since they are dependent upon the same interconnect physical dimensions. Interconnect resistance is maximized for small and narrow interconnects and interconnect capacitance is maximized for thick and wide interconnects. Therefore, assuming the worst-case values for both interconnect resistance and interconnect capacitance is

pessimistic, since both cannot be maximized (or minimized for hold-time analysis) in the same interconnect corner [50]. Therefore, an accurate timing-analysis framework must account for the correlations between the interconnect resistance and capacitance. In [42, 43], it was shown that the best-case and the worst-case interconnect corners, which result in minimum and maximum interconnect delays, do not always occur at the extremes of the interconnect dimensions. The interconnect corners were computed by exhaustively searching for optimal interconnect parameters within their range of variations. In [29], the worst-case interconnect corners were computed considering variability in the interconnect dimensions and the driver strengths. In [8, 52] statistical analysis of the interconnect delay was performed using model order reduction techniques.

The variations in the interconnect coupling capacitance (C_c) and the interconnect ground capacitance (C_g) are strongly correlated, and an increase in the interconnect dimensions always leads to an increase in the magnitude of both C_c and C_g. The victim-stage delay always increases with an increase in the victim ground capacitance. However, when coupling noise is injected on the victim due to the switching transition of an aggressor, the victim-stage delay could actually *decrease* with an *increase* in the interconnect coupling capacitance. This counterintuitive scenario occurs when the aggressor-victim nets are switching in the same direction and the interconnects have substantial coupling capacitance. Hence, an increase in the coupling capacitance leads to an increase in the magnitude of the delay noise, which results in a lower victim-stage delay. Therefore, for such cases, the best-case interconnect corner, resulting in the minimum victim-stage delay, occurs when the coupling capacitances have the maximum value.

However, prior approaches [42, 43] compute the interconnect corners under the

assumption that the aggressor nets are not switching and there is no coupling noise injected on the victim net. Hence, prior approaches which do not model the impact of the coupling noise on the victim-stage delay could result in erroneous analysis. Therefore, the interconnect corners reported by these approaches could be significantly different from the *true* interconnect corner. In this work, we propose to compute the true interconnect corners for the victim by accounting for the effects of coupling noise due to simultaneous switching of aggressors. In this work, we use the Elmore-delay metric to efficiently search for the true interconnect corners of the victim stage considering delay noise. We then show experimental results to verify the accuracy and effectiveness of our proposed approach, and demonstrate that the traditional approaches of computing the interconnect corners could lead to errors of up to 60% in the victim-stage delay.

The remainder of the chapter is organized as follows: In Section 7.1, we analyze the problem of finding the best-case interconnect corners in greater detail. In Section 7.2, we present an approach to compute the interconnect corners by also considering the effects of delay noise. In Section 7.3, we show experimental results that confirm the effectiveness of our approach, and in Section 7.4, we summarize the chapter.

7.1 Problem Description

In this section we analyze the problem of computing the best-case interconnect corners considering process variations in devices and interconnects. Figure 7.1 shows the cross-section of the interconnect layers, where W and T represent the lateral width and the vertical thickness of an interconnects, respectively. The spacing between adjacent interconnect nets in the same layer is given by S and the inter-layer dielectric separation (ILD thickness) is given by H. Due to the variations in ox-

ide deposition and the Chemical Mechanical Planarization (CMP) process, the ILD thickness (H) can vary substantially (e.g., $H \in [H_{min}, H_{max}]$) within a die due to layout-pattern dependencies [16]. Also, variability of the photolithography equipment and the random dishing effects can cause significant variations in the width and thickness of the interconnect (e.g., $W \in [W_{min}, W_{max}]$ and $T \in [T_{min}, T_{max}]$). Note that the variations in the interconnect parameters (W, T and H) can be assumed to be mutually uncorrelated, since they are independently caused by different steps within the manufacturing process. With the exception of variations in W and S, which are assumed to be perfectly negatively correlated because an increase in the interconnect width would lead to an equivalent decrease in the lateral spacing.

The worst-case and the best-case interconnect corners can be obtained by performing exhaustive simulations at different interconnect corners. Often, the interconnect delay is dominated by either the interconnect resistance or the interconnect capacitance. To reduce the search space, worst-case process-corners are often heuristically chosen such that they maximize either the interconnect capacitance or the

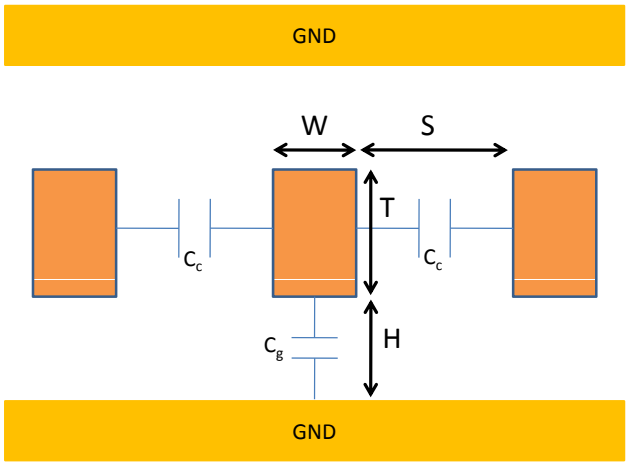


Figure 7.1: A cross-section of interconnect metal layers.

interconnect resistance. Note that the interconnect resistance is maximized for thin and narrow interconnects (W_{min}, T_{min}). Conversely, the interconnect capacitance is maximized for thick and tall interconnects (W_{max}, T_{max}). Similarly the *best-case* interconnect corner, required for hold-time analysis, is assumed to coincide with the interconnect corner having either the minimum interconnect resistance (W_{max}, T_{max}) or the minimum interconnect capacitance (W_{min}, T_{min}). However, it must be noted that the above approach is only a heuristic and could result in optimistic timing analysis. Also, while performing timing analysis of the chip, the interconnect corners have to be combined with the Process-Voltages-Temperature (PVT) corners of a design. Hence, the approach presented above essentially *doubles* the total number of corners that have to be analyzed for a design.

It was shown in [42] that the above heuristic often computes the worst-case interconnect corners accurately. However, the best-case interconnect corner does not necessarily coincide with the minimum interconnect resistance and capacitance corners. Therefore, for hold-time analysis, which requires the best-case interconnect corners, the above heuristic could result in an optimistic analysis. Several simulation-based approaches have been proposed [29, 42, 43] to search for the best-case interconnect corner, which results in the minimum interconnect delay. However, in all previous approaches the best-case corner is computed under the assumption that the aggressor nets are not switching. In other words, the victim-stage delay is simulated by assuming that the floating coupling capacitances are grounded. With coupling capacitance accounting for more than 85% of the total wiring capacitance [69], one cannot ignore the delay noise on the victim net which occurs due to the simultaneous switching of the aggressor nets. Hence, the *true* best-case interconnect corner for the victim must incorporate the change in victim-stage delay due to delay noise.

7.1.1 Non-Monotonic Victim-Stage Delay

The victim-stage delay exhibits a nonmonotonic relationship with the interconnect width (W). At very small interconnect widths, the victim interconnect has a significant amount of resistance. Therefore, in this region, the victim-stage delay is dominated by the interconnect resistance and is very sensitive to the variations in the interconnect resistance. Hence, at smaller interconnect widths, a *decrease* in the interconnect width leads to an increase in the magnitude of the interconnect resistance and consequently an *increase* in the victim-stage delay. At larger interconnect widths, the interconnect has a significant amount of capacitance, and the victim-stage delay is dominated by the interconnect capacitance. A further increase in the interconnect width would lead to an increase in the interconnect coupling and ground capacitances. Therefore, at larger interconnect widths, the victim-stage delay is directly proportional to W. Hence, the victim-stage delay exhibits a nonmonotonic relationship with respect to its interconnect width. Consequently, the victim-stage delay is minimized at an intermediate interconnect width ($W_{opt} = 0.37\mu m$ in Figure 7.2). A similar nonmonotonic relationship can be observed when we obtain the victim-stage delay as a function of the interconnect thickness T.

As we know, process variations could lead to variability in the interconnect dimensions. In Figure 7.2, the region to the right of W_{opt} corresponds to the case when the interconnect delay is capacitance dominated. Therefore, in this region, the victim-stage delay is minimized at $W = W_{min}$, since the range of feasible variations [W_{min}, W_{max}] lies to the right of W_{opt}. Similarly, for resistance dominated interconnects, the region of feasible variations lie to the left of W_{opt}, and the victim-stage delay is minimized at $W = W_{max}$. However, there can also be the cases when W_{opt} lies inside the variation range [W_{min}, W_{max}]. It would be erroneous to assume, for all

such cases that the best-case interconnect corner coincides with either W_{min} or W_{max}. Therefore, traditional approaches which maximize only the interconnect resistance or capacitance could potentially miss the best-case interconnect corner for all such cases.

7.1.2 Dependence on Coupling Capacitance

We illustrate with an example the dependence of the victim-stage delay with the coupling capacitances C_c. Figure 7.3 illustrates coupled aggressor-victim nets with falling transitions at the inputs of the aggressor-victim drivers. Suppose the aggressor driver is lightly loaded relative to the victim, and we obtain relatively a faster aggressor-output transition compared to the victim-output transition. Therefore, the aggressor transition is completed within the time interval Δt during which the victim transition rises only up to the $0.5V_{DD}$ voltage level. The charges transferred through the coupling capacitance C_c during the time interval Δt depend only on

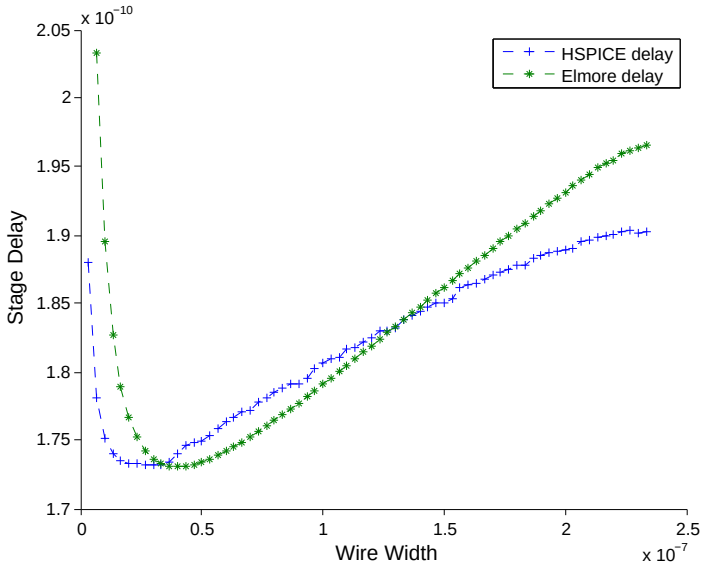


Figure 7.2: Elmore delay versus HSPICE-simulated delay as a function of the interconnect width (W).

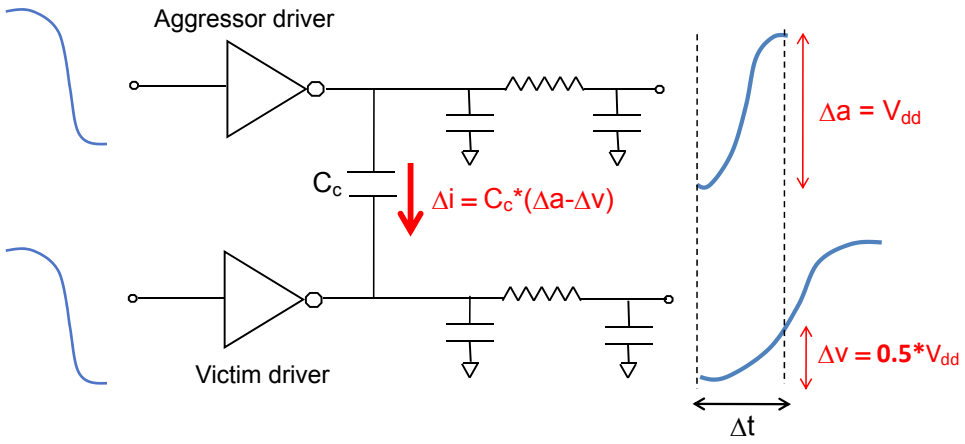


Figure 7.3: Switching of aggressor-victim nets in the same direction.

the magnitude of the coupling capacitance and the voltages at the extremes of the coupling capacitance. The total current (Q) flowing into the victim node through the coupling capacitance C_c can be expressed as,

$$Q(\Delta t) = \int_0^{\Delta t} i_{C_c} \cdot dt = C_c * (\Delta a - \Delta v), \tag{7.1}$$

where Δa and Δv represent the change in the aggressor-victim output voltages within the time interval Δt, respectively. Hence, if the aggressor output transition rises faster than the victim (i.e., $\Delta a > \Delta v$), then it leads to a positive injection of current into the victim node which results in a speed up of the victim transition as compared to a scenario where we have a quiet aggressor. It follows from Equation 7.1 that the magnitude of the charge injected through the coupling capacitance is directly proportional to the magnitude of the coupling capacitance. Hence, if the aggressor transition is faster than the victim transition, then an increase in the coupling capacitance will lead to a further speed up of the victim transition due to an increase in the amount of the total charge injected on the victim net.

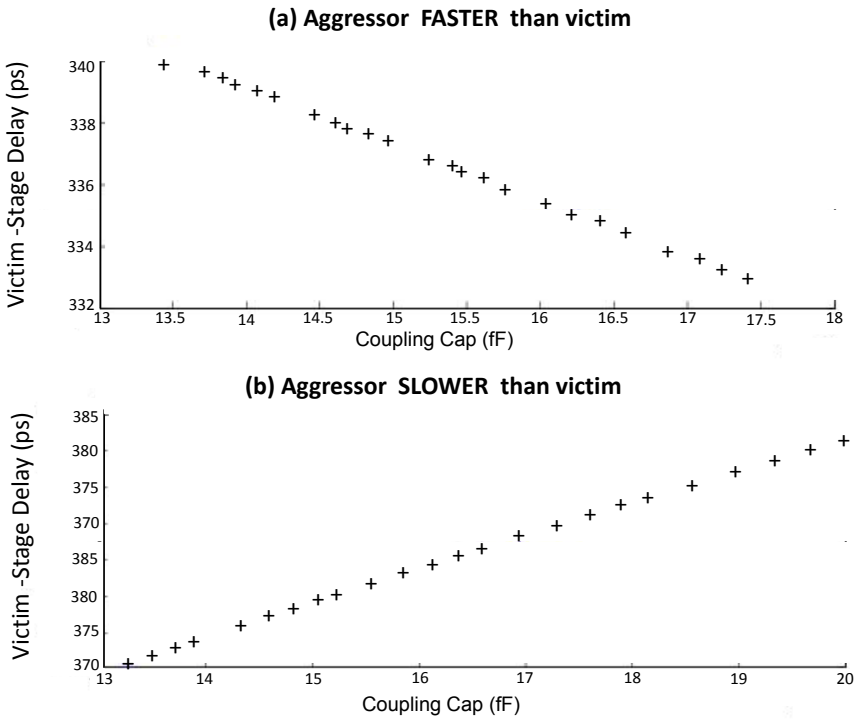


Figure 7.4: Dependence of the victim-stage delay on the coupling capacitance.

The above relationship is verified in Figure 7.4 where we plot the victim-stage delay as a function of the coupling capacitance C_c. The simulations were performed using the HSPICE simulator with the RC interconnect parameters extracted in the $65nm$ process-technology node. The coupling capacitance was varied by changing the lateral spacing between the aggressor and victim nets. As discussed earlier, we find that the victim-stage delay *decreases* with an increase in the coupling capacitance C_c (in Figure 7.4a) provided the aggressor transition is faster than the victim. A similar experiment was performed by changing the relative loadings of the aggressor-victim drivers such that the victim transition is faster than the aggressor transition (i.e., $\Delta v > \Delta a$). In this case, there is a net outflow of current from the victim node to the aggressor node through the coupling capacitance. An increase in the coupling capacitance would lead to an increase in the magnitude of the total current flowing out through the coupling capacitance, and finally result in a further slow down of the victim transition. Hence, we observe that the victim-stage delay *increases* with

an increase in the magnitude of the coupling capacitance (in Figure 7.4b) given that the aggressor transition is slower than the victim transition.

Therefore, depending on the relative rate of the aggressor-victim transitions, the victim-stage delay is a nonmonotonic function of the victim coupling capacitance (C_c). In comparison, the victim-stage delay is always a monotonic function of the victim ground capacitance (C_g). However, previous approaches do not model the nonmonotonicity in the victim-stage delay which occurs due to coupling noise. Instead, the best-case interconnect corner is obtained by using the total interconnect capacitance ($C_g + C_c$) and assuming that none of the aggressors are switching. In the results section of this chapter, we show that the above assumption could lead to significant errors while computing the best-case interconnect corner.

7.1.3 Correlations between Coupling and Ground Capacitance

The values of the interconnect coupling capacitance C_c and the interconnect ground capacitance C_g are strongly correlated with each other, since they depend on the same interconnect physical dimensions. An increase in the size (W, T) of an interconnect leads to an increase in the magnitude of both C_c and C_g. When the aggressor nets are not switching and there is no coupling noise, the victim-stage delay always increases with an increase in total interconnect capacitance ($C_g + C_c$). However, with coupling noise, the victim-stage delay can either increase or decrease with an increase in the coupling capacitance (C_c). Therefore, it is difficult to predict a priori, whether an increase in the total interconnect capacitance ($C_g + C_c$) would lead to an increase in the victim-stage delay when there is coupling noise.

In previous approaches, the best-case interconnect corner is obtained by using the total interconnect capacitance ($C_g + C_c$) and assuming no switching of aggressors. With coupling capacitance accounting for more than 85% of the total wiring capaci-

tance [69], the best-case interconnect corners computed using the above assumption could result in significant errors.

7.2 Proposed Approach

In the previous section we saw that the victim-stage delay can either increase or decrease with an increase in the interconnect coupling capacitance. Hence, even for a capacitance-dominated interconnect, the best-case interconnect corner may not coincide with the minimum interconnect-capacitance corner. A *brute-force* approach to compute the best-case interconnect corner would be to sweep all the interconnect parameters (W, S, T, H) within their range of variations (e.g., $W \in [W_{min}, W_{max}]$), and select the interconnect corner that results in the minimum stage delay. It can however be noted that the interconnect capacitance is inversely proportional to the ILD thickness H and the interconnect resistance is independent of H. Consequently, the interconnect delay is always minimized at H_{max} [43]. We also know that the variations in W and S are perfectly negatively correlated. Therefore, the brute-force approach to locate the best-case interconnect corner would require a two-dimensional sweep in the parameters W and T. However, the total number of simulations required in a brute-force approach can be very large and computationally very expensive.

In this section, we present an approach to find the best-case interconnect corner by using the Elmore-delay metric [59]. The Elmore delay of an RC tree provides a dominant pole approximation of the interconnect delay and can be computed very efficiently as follows :

$$Elmore\ Delay = \sum_{i=0:N} \left(R_i \sum_{k=i:N} C_k \right), \tag{7.2}$$

where N refers to the number of nodes in the RC tree. If the root node of the RC tree is connected to the driver-output resistance R_0, then the Elmore delay provides a first-order approximation of the actual stage delay.

It must be noted that Elmore delay is only defined for an RC tree where all the node capacitances have a terminal connected to the ground. In order to compute the Elmore delay of the victim stage, the coupling capacitance of the victim must be decoupled from the aggressor node. Therefore, we must find an equivalent Miller-coupling capacitance that models the change in the aggressor voltage (see Figure 7.5). If the aggressor-victim nets switch in the same direction, then the effective Miller-coupling capacitance can be written as

$$C_c^{eff} = C_c \left(1 - \frac{\Delta a}{\Delta v}\right) \tag{7.3}$$

where Δa and Δv are the changes in the voltages across the coupling capacitances at the victim and the aggressor nodes. In [3, 22] techniques were proposed to accurately compute the Miller-capacitance C_c^{eff} by iteratively updating the aggressor-victim waveforms and refining the values of Δa and Δv. The C_c^{eff} can be efficiently computed by approximating the aggressor-victim waveforms with ramps. Once the coupling capacitance is decoupled from the aggressor node, the Elmore delay of the victim can be computed using the Miller-capacitance (as shown in Equation 7.2).

While the Elmore delay may not always be very accurate, its usefulness lies in the fact that it can be computed very efficiently, and it captures the relationship between the actual stage delay and the interconnect parameters with reasonable accuracy. In order to validate the above claim, we plot in Figure 7.2 the victim-stage delay as a function of the victim interconnect width W. The simulations were performed

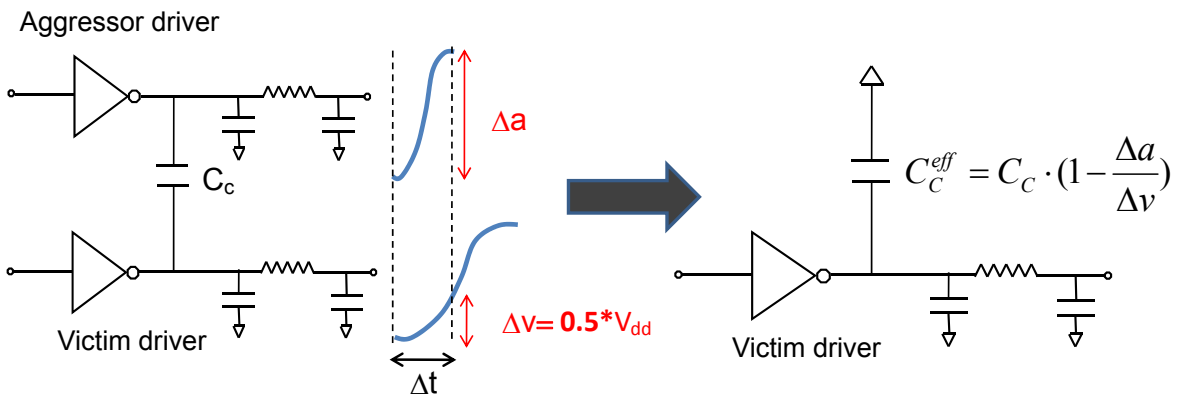


Figure 7.5: Victim-stage analysis with Miller-coupling capacitance.

for the circuit shown in Figure 7.3 in the $65nm$ technology node with the following interconnect parameters, $T = 0.35\mu m,\ S = 0.25\mu m,\ H = 0.2\mu m$ and $L = 150\mu m$. The interconnect resistance and capacitances (per unit length) were obtained as functions of the interconnect parameters using the following equations [74],

$$R = \frac{\rho}{W \cdot T},$$

$$\frac{C_g}{\epsilon_{ox}} = 1.171 \left(\frac{S}{S + 1.51H}\right)^{0.7642} \cdot \left(\frac{T}{T + 4.532H}\right)^{0.1204} + \frac{W}{H} + 2.217 \left(\frac{S}{S + 0.702H}\right)^{3.193},$$

$$\frac{C_c}{\epsilon_{ox}} = 1.158 \left(\frac{W}{W + 1.874S}\right)^{0.1612} \cdot \left(\frac{H}{H + .9801S}\right)^{1.179} + 1.144 \frac{T}{S} \left(\frac{H}{H + 2.059S}\right)^{0.0944} + 0.7428 \left(\frac{W}{W + 1.592S}\right)^{1.144},$$

$$\rho = 2.2\ \mu\Omega - cm\ \ ,\ \ \epsilon_{ox} = 3.9 * 8.85 * 10^{-14} F/cm. \tag{7.4}$$

The *convex* relationship of the victim-stage delay with respect to the interconnect

width W is illustrated in Figure 7.2. In this plot, the victim-stage delay was minimized at an intermediate width of $W_{opt} = 0.37\mu m$. We also plot in Figure 7.2 the Elmore delay of the victim as a function of interconnect width W. It can be seen that the Elmore delay nicely captures the fidelity of the actual stage delay with respect to W. In the above experiment, the Elmore delay was minimized at $W = 0.4\mu m$. Since the Elmore delay tracks the actual stage delay very well and can be computed very efficiently, we use it to compute the best-case interconnect corner.

A brute-force approach would require a 2-D sweep of the parameters W and T within their range of variations (e.g., $W \in [W_{min}, W_{max}]$) and the computation of the Elmore delay at every point. However, one can leverage the convex nature of the relationship between the stage delay and the interconnect width W, and easily evaluate whether the interconnect delay is minimized at the boundaries W_{min} or W_{max}. We first compute the sensitivities of the Elmore delay with respect to the width W at the boundaries W_{min} and W_{max}, respectively. If both sensitivities are positive, then it implies that the range of feasible interconnect widths [W_{min}, W_{max}] lies to the right of the minimum delay width W_{opt}. For all such cases, using the convex property, the feasible interconnect width at which the stage delay is minimized at $W = W_{min}$. Similarly, if the sensitivities of the Elmore delay at the interconnect-width boundaries are negative, then it implies that the range of feasible interconnect widths lie to the left of W_{opt}. Hence, for such cases, the stage delay is minimized at $W = W_{max}$. Finally, for cases where the sensitivities at the boundaries differ in sign, it can be inferred that W_{opt} lies within the range [W_{min}, W_{max}]. A gradient-based approach such as Newton-Raphson method could be used to search for W_{opt}. A similar approach could be used to compute an optimal interconnect thickness T_{opt}. Since the interconnect corners are evaluated using the Elmore delay,

the overall algorithm to find the best-case interconnect corner is runtime efficient. In the results section of this chapter, we will show the accuracy of the interconnect corners computed using the above approach.

7.3 Experimental Results

In this section, we will show experimental results that verify the accuracy and effectiveness of our proposed approach for computing the best-case interconnect corner. We first show that significant errors can be introduced when the best-case corner is computed under the assumption that the aggressor nets are not switching. Experimental results confirm the fact that when coupling noise is not accounted, one could potentially miss the true interconnect corners leading to optimistic delay analysis.

The experiments were performed in the $65nm$ technology node using the fully-coupled aggressor-victim circuit shown in Figure 7.3. The aggressor-victim interconnects have the following dimensions $W = 0.14\mu m$, $H = 0.2\mu m$ and $T = 0.35\mu m$ as suggested in [2]. A $3\sigma/\mu$ intra-die variability of 30% was assumed for the interconnect parameters W and T. Multiple instances of the aggressor-victim coupled circuit were created by changing the parameters of the drivers and the interconnects. A total of 32 different circuits were created by permutating the following parameter values: the victim-input slew rates $(10ps, 200ps)$, the aggressor-victim driver strengths $(2X, 12X)$ with respect to minimum-sized drivers, the interconnect lengths $(50\mu m, 200\mu m)$ and the aggressor-victim interconnect spacing S $(0.14\mu m, 0.45\mu m)$.

In order to confirm the importance of considering coupling noise in our analysis, we find the best-case interconnect corners under the following two scenarios (1) the aggressor net is not switching, and (2) the aggressor net is switching in the same direction as the victim net. In both cases, the best-case interconnect corner was

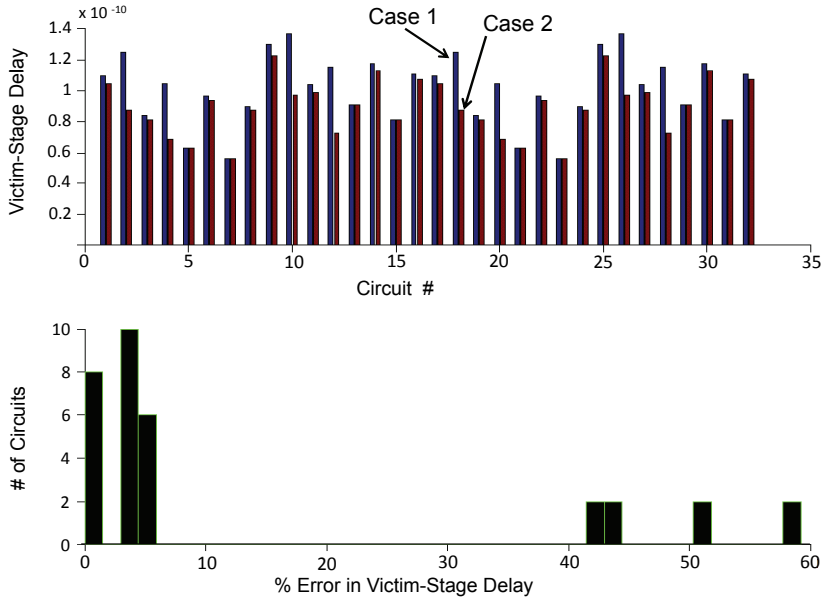


Figure 7.6: Victim-stage delay obtained at different interconnect corners.

obtained by exhaustively sweeping the parameters W and T with a discretization step size of $1nm$. At each step, the corresponding variation in the interconnect capacitances and resistances was obtained by using Equation 7.4, and the victim-stage delay was obtained using the HSPICE circuit simulator. In Case 1, the input of the aggressor driver is assumed to be grounded. In Case 2, the aggressor transition was enumerated with a discretization step of $2ps$, and an optimal aggressor alignment was computed such that it resulted in the minimum victim-stage delay.

It is apparent that the interconnect corner obtained in Case 2 is the true (golden) interconnect corner, since it results in the minimum victim-stage delay with coupling noise. In comparison, the interconnect corner obtained in Case 1 results in the minimum victim-stage delay when there is *no* coupling noise. Hence, if the interconnect corners obtained in both cases differ appreciably, then the interconnect corners obtained in Case 1 could lead to optimistic analysis with coupling noise. Figure 7.6 compares the victim-stage delays obtained with coupling noise at both interconnect

corners for each of the 32 circuits.

It can be observed that the victim-stage delay reported in Case 1 has large errors for several of the circuits. When these circuits were further analyzed, it was seen that the interconnect corner obtained for Case 1 coincided with the minimum sized interconnects such that the total interconnect capacitance was minimized. In contrast, the golden corner obtained in Case 2 coincided with the tallest interconnects such that the coupling capacitance was maximized. Hence, for these circuits where coupling noise contributes significantly to the victim-stage delay, maximizing the coupling capacitance results in the minimum victim-stage delay. A histogram of the percentage error in victim-stage delay with respect to the golden victim-stage delay for each circuit is also shown in Figure 7.6. An error of up to 60% is observed in the victim-stage delay when coupling noise is not accounted for in the computation of the best-case interconnect corner.

A third set of interconnect corners were constructed for each circuit by instead using the Elmore delay as a proxy for the victim-stage delay. A two-dimensional brute-force sweep was performed in the W and T parameter space, and the Elmore delay was computed at each step using Miller-coupling capacitance. Finally the interconnect corner that resulted in the smallest Elmore delay was reported as the best-case interconnect corner. Figure 7.7 illustrates that the proposed interconnect corner matches closely with the golden interconnect corner. The maximum error in the victim-stage delay was less than 3% across all circuits. Therefore, Elmore delay captures the fidelity of the victim-stage delay with respect to the interconnect parameters and can be used to accurately compute the best-case interconnect corner.

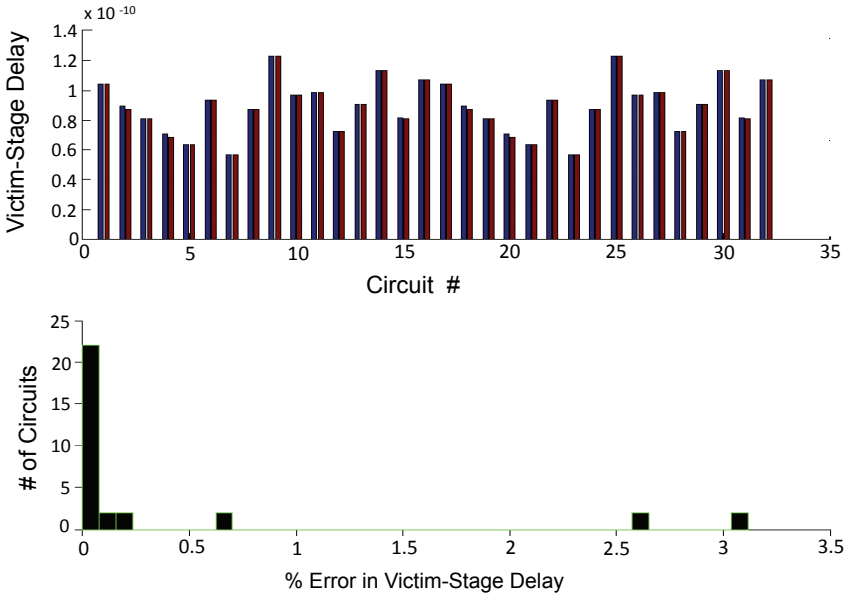


Figure 7.7: Victim-stage delay at the proposed interconnect corner versus golden corner.

7.3.1 Results on Benchmark Circuits

It can be noted that the hold-time violations are typically caused by fast paths between pipeline stages with single or no logic gates. We saw earlier that the interconnect corners computed by assuming no coupling noise could lead to errors of up to 60% in the stage delays when there is a significant amount of coupling capacitance. In comparison, the LGSynth91 benchmark circuits [1] are all combinatorial circuits, and all the fast paths reported in Table 7.1 have a circuit depth ranging between two to eight stages. Therefore, in most cases, they are not representative of the fast paths that cause hold-time violations in a design. However, for the sake of completeness, we show experimental results to verify the effectiveness of our proposed approach on LGSynth91 benchmark circuits . These benchmark circuits were synthesized in the 130nm process technology. The place-and-route of the benchmark circuits was performed using the Cadence Silicon Ensemble tool with up to three metal layers and

Table 7.1: Fast-path delays for LGSynth91 benchmark circuits.

ckt	Nominal Path Delay (ps)	Path Delay with Xtalk (Nominal)	Path Delay with Xtalk (MIN Capacitance)	Path Delay with Xtalk (Proposed Corner)	%Error in Path Delay Noise
i1	60.7	49.7	48.9	47.8	9.32
i2	83.4	73.7	72.2	70.5	15.17
i3	42.97	37.1	36.3	35.5	11.99
i4	49.3	40.5	40.3	40.3	0
i5	42.8	39.4	38.2	38.2	0
i6	64.2	49.4	48.2	46.7	9.37
i7	61.5	48.1	46.8	45.4	9.52
i8	137.5	117.7	117.0	115.4	7.80
i9	180.1	167.2	166.7	166.7	0
i10	289.8	285.4	282.6	282.6	0

a target floorplan utilization of 80%. The Mentor Graphics Calibre tool was used to extract the parasitic coupling and ground capacitances for the nets in the design. A noise-analysis engine was implemented in the C++ programming language and the circuit delay was computed using industrial timing libraries.

In Table 7.1, we list the fast-path delays obtained at different interconnect corners for each of the LGSynth91 benchmark circuits. A $3\sigma/\mu$ intra-die variability of 30% was assumed for the interconnect parameters W and T. In Column 2, we report the fast-path delays at the nominal interconnect corner by assuming that all aggressors coupled to the path are not switching. In Column 3, we report the fast-path delays at the nominal interconnect corner by accounting for the speed-up due to coupling noise. Next, we construct an interconnect corner with minimum total interconnect capacitances by using the smallest sized interconnects (W_{min}, T_{min}). In our experiments, most of the interconnects have negligible interconnect resistances and are capacitance dominated. Therefore, the smallest sized interconnects with the least interconnect capacitance would lead to minimum path delays provided there is no coupling noise. In Column 4, we report the fast-path delays with coupling noise computed at the minimum-capacitance interconnect corner. Finally, using the proposed

approach, we find the best-case interconnect corner for each victim net on the fast paths. In Column 5, we report the fast-path delays obtained at the interconnect corner constructed above. One can observe that the best-case interconnect corner does not always coincide with the minimum-capacitance interconnect corners. Therefore, the path delays reported in Column 5 are always lesser than or equal to that obtained at the minimum-capacitance interconnect corner. In the final column, we report the percentage error in the path delay noise obtained at the minimum capacitance corner. The minimum-capacitance corner results in the minimum path delay for a few circuits (e.g., $i10$). However, for other circuits (e.g., $i2$), the minimum-capacitance interconnect corner leads to an error of up to 15% in the path delay noise.

7.4 Summary

Process variations in the interconnect capacitance and resistance could lead to significant uncertainty in the interconnect delays. In this chapter, we proposed a new method to compute the best-case interconnect corner considering coupling noise due to simultaneous switching of aggressors. In prior approaches, the best-case interconnect corners were computed under the assumption that the aggressor nets are not switching and no coupling-noise pulse is injected on the victim net. In this chapter, we first show that the interconnect corners obtained under such assumptions could be significantly different from the best-case interconnect corner and could therefore result in optimistic delay analysis. We used the Elmore-delay metric to efficiently search for the best-case interconnect corner of the victim stage considering delay noise. Experimental results verified the accuracy and effectiveness of our proposed approach, and demonstrated that the traditional approaches of computing the interconnect corners could lead to errors of up to 60% on a net by net basis.

CHAPTER VIII

Conclusion and Future Work

We focused on the analysis and modeling of crosstalk noise for VLSI chips in the nanometer process technology. Several approaches were proposed to solve a few key problems focusing on the analysis and accurate modeling of crosstalk noise, and also circuit optimization considering crosstalk noise.

In this thesis, we presented an analytical result that would obviate the need to search for the worst-case victim transition and thereby simplifying the aggressor-victim alignment problem significantly. We also proposed a heuristic approach to compute the worst-case aggressor alignment that maximizes the victim receiver-output arrival time with nonlinear-driver models. Increasing process variation in the nanometer process technology motivated the use of SSTA tools for timing verification. Process variations cause variability in the aggressor-victim alignment which leads to variability in delay noise. We proposed an approach to represent the delay-noise distribution in canonical form, which could be easily integrated into a standard statistical timing analysis tool. We also show that interconnect corners obtained without incorporating the impact of coupling noise could lead to significant errors. In this thesis, we proposed a new technique to compute the best-case interconnect corner considering the impact of coupling noise. Worst-case noise analysis can be

very pessimistic, and consequently there is a need for more accurate and less pessimistic noise-analysis approaches. In this thesis, we proposed a statistical framework for estimating the confidence bounds on the path delay-noise distribution. We also developed novel algorithms to identify the set of *top-k* aggressors in the circuit, which could then be fixed to optimally reduce the circuit delay noise.

Possible future work for the research problems that were addressed in this thesis are as follows:

- In this thesis, while modeling the delay-noise distribution, we accounted for only gate-length variations as they are the dominant sources of process variations. A more accurate analysis could be performed by accounting for other sources of variations such as the victim-slew variations and the aggressor-victim interconnect variations.

- Extend the proposed approach of computing the set of the top-k aggressors for nonlinear-driver models.

- The latest-victim alignment result was established by assuming lumped interconnect-capacitance model. It would be interesting to see whether a similar result can be proved for the case when then aggressor-victim interconnects are represented by distributed coupled RC loads.

- More accurate interconnect corners can be computed by using higher-order delay metrics (e.g., D2M) instead of using the first-order Elmore-delay metric.

APPENDICES

APPENDIX A

Inequality Between Victim Load Currents

In this section, we prove the following inequality between instantaneous load currents at crossover time τ_v,

$$i^l_{r,v}(t) \; > \; i^e_{r,v}(t) \;\; |_{t=\tau_v} \; ,$$

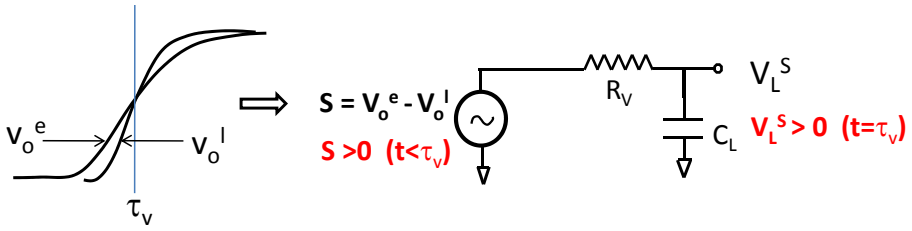
where $i^e_{r,v}(t)$ and $i^l_{r,v}(t)$ are the currents flowing into the victim load ($R_v C_L$) corresponding to the early ($v^e_o(t)$) and late $\left(v^l_o(t)\right)$ victim driver output transitions, respectively. The current $i^e_{r,v}(t)$ is a function of the voltage differences across the terminals of the resistance R_v, i.e.,

$$i^e_{r,v}(t) \; = \frac{v^e_o(t) \; - \; v^e_L(t)}{R_v},$$

where $v^e_L(t)$ and $v^l_L(t)$ are the early and late voltages across the load capacitance C_L. Subtracting the instantaneous early and late victim load currents at the crossover time τ_v, we obtain,

$$i^l_{r,v}(t) - i^e_{r,v}(t) \;\; |_{t=\tau_v} = \frac{\{v^l_o(\tau_v) - v^l_L(\tau_v)\} \; - \; \{v^e_o(\tau_v) - v^e_L(\tau_v)\}}{R_v}.$$

However, by definition of crossover time, the early and late victim waveforms have the same value at crossover time τ_v $\left(i.e.,\; v^l_o(\tau_v) = v^e_o(\tau_v)\right)$. Hence, the difference



between instantaneous load currents is obtained as,

$$i^l_{r,v}(t) - i^e_{r,v}(t) \mid_{t=\tau_v} = \frac{v^e_L(\tau_v) - v^l_L(\tau_v)}{R_v}.$$

Using linear-superposition principle, we will prove that the term on the right hand side of the above equation is always positive. The victim load (R_vC_L) being charged by a voltage source $S(t)$,

$$S(t) \;=\; v^e_o(t) \;-\; v^l_o(t).$$

obtained by subtracting the late victim $v^l_o(t)$ from the early victim $v^e_o(t)$. From definition, τ_v is the first time when the victim waveforms cross each other,

$$S(t) \;\geq\; 0, \quad \forall t \;\leq\; \tau_v.$$

During the time $t < \tau_v$, the load (R_vC_L) is being charged by a nonnegative voltage source $S(t)$. Therefore, at crossover time τ_v, the load capacitance will have accumulated a net positive charge. Therefore,

$$V^S_L(\tau_v) \;>\; 0.$$

Using linear-superposition principle, we can rewrite the above equation as follows,

$$v^e_L(\tau_v) - v^l_L(\tau_v) \;>\; 0.$$

Finally, using the above inequality, we establish the inequality between load currents,

$$i^{l}_{r,v}(t) - i^{e}_{r,v}(t) \; > \; 0.$$

Note that the above inequality between the load current also holds when the victim drives a distributed RC load.

APPENDIX B

First and Second Moments of Delay-Noise Distribution

In this Appendix, we derive the first and second moments of the delay-noise distribution for first quadratic piece of the DCC (i.e., $a_1 \cdot s^2 + b_1 \cdot s + c_1$), and note that the derivation of the moments for the second piece is analogous.

While computing the expectation of f_y, we first perform a transformation of the variable s to z,

$$z = \frac{1}{2a_1} \cdot \sqrt{b_1^2 - 4 \cdot a_1 \cdot (c_1 - s)}.$$

The limits of the integration become

$$z_0 = \frac{1}{2a_1} \cdot \sqrt{b_1^2 - 4 \cdot a_1 \cdot (c_1 - d_{max})},$$
$$z_1 = \frac{1}{2a_1} \cdot \sqrt{b_1^2 - 4 \cdot a_1 \cdot c_1},$$

where d_{max} is the peak delay noise in DCC. The first moment of the first term in the PDF of delay noise is given by

$$M_1 = I_1(z_1) - I_1(z_0),$$

where I_1 is the indefinite integral given by the following,

$$I_1(z) = \frac{1}{2\sqrt{2\pi}} exp\left(-\frac{1}{8s^2}\left(4z^2 + \frac{K_1^2}{a_1}\right)\right) \cdot \Bigg(\quad (K_1 + 2a_1 z) \cdot s \cdot \exp\left(-\frac{K_1 z}{2a_1 s^2}\right) +$$

$$2\sqrt{2\pi} exp\left(-\frac{K_1^2 + 4a_1^2 z^2}{8a_1^2 s^2}\right) \cdot \left(c_1 + b_1\mu + a_1(s^2 + \mu^2)\right) \cdot erf\left(\frac{2a_1 z - K_1}{2\sqrt{2}a_1 s}\right) \quad \Bigg),$$

$$K_1 = b_1 + 2a_1\mu.$$

Similarly, the second moment of the delay-noise distribution can be computed as follows,

$$M_2 = I_2(z_1) - I_2(z_0),$$

where $I_2(z)$ is the indefinite integral given by the following,

$$I_2(z) = \frac{1}{8a_1\sqrt{2\pi}} exp\left(-\frac{1}{8s^2}\left(4z^2 + \frac{K_1^2}{a_1}\right)\right) \cdot \Bigg(\quad K_2 \cdot s \cdot \exp\left(-\frac{K_1 z}{2a_1 s^2}\right) +$$

$$4a_1\sqrt{2\pi} \cdot K_3 \cdot exp\left(-\frac{K_1^2 + 4a_1^2 z^2}{8a_1^2 s^2}\right) \cdot erf\left(\frac{2a_1 z - K_1}{2\sqrt{2}a_1 s}\right) \quad \Bigg),$$

$$K_1 = b_1 + 2a_1\mu,$$

$$K_2 = -b_1^3 + 2a_1 b_1^2(\mu - z) + 4a_1 b_1(2c_1 + a_1(5s^2 + 3\mu^2 + 2\mu z + z^2))$$

$$+8a_1^2\left(2c_1(\mu + z) + a_1(\mu^3 + \mu^2 z + \mu z^2 + z^3 + s^2(5\mu + 3z))\right),$$

$$K_3 = c_1^2 + b_1^2(s^2 + \mu^2) + 2a_1 b_1\mu(3s^2 + \mu^2)$$

$$+a_1^2(3s^4 + 6s^2\mu^2 + \mu^4) + 2c_1(b_1\mu + a_1(s^2 + \mu^2)).$$

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] http://www.cbl.ncsu.edu:16080/benchmarks/LGSynth91/cmlexamples

[2] Predictive technology model., http://www.eas.asu.edu/p̃tm.

[3] S. Abbaspour and M. Pedram. Gate delay calculation considering the crosstalk capacitances. In *Proc. ASP-DAC*, pages 852–857, 2004.

[4] A. Agarwal, D. Blaauw, and V. Zolotov. Statistical timing analysis using bounds and selective enumeration. In *IEEE Trans. on CAD*, pages 1243–1260, 2003.

[5] A. Agarwal, D. Blaauw, V. Zolotov, S. Sundareswaran, M. Zhou, K. Gala, and R. Panda. Statistical delay computation considering spatial correlations. In *Proc. ASP-DAC*, pages 271–276, 2003.

[6] K. Agarwal, M. Agarwal, D. Sylvester, and D. Blaauw. Statistical interconnect metrics for physical design optimization. In *IEEE Trans. on CAD*, pages 1273–1288, 2006.

[7] K. Agarwal, Y. Cao, T. Sato, D. Sylvester, and C. Hu. Efficient generation of delay change curves for noise-aware static timing analysis. In *Proc. ASPDAC*, pages 77–84, 2002.

[8] K. Agarwal, D. Sylvester, D. Blaauw, F. Liu, S. Nassif, and S. Vrudhula. Variational delay metrics for interconnect timing analysis. In *Proc. DAC*, pages 381–384, 2004.

[9] C. Amin, C. Kashyap, N. Menezes, K. Kilpak, and E. Chiprout. A multiport current source model for multiple-input switching effects in cmos library cells. In *Proc. DAC*, pages 247–252, 2006.

[10] R. Arunachalam, R. D. Blanton, and L. T. Pileggi. False coupling interactions in static timing analysis. In *Proc. DAC*, pages 726–731, 2001.

[11] R. Arunachalam, F. Dartu, and L. T. Pileggi. Cmos gate delay models for general rlc loading. In *Proc. ICCD*, pages 224–229, 1997.

[12] R. Arunachalam, K. Rajagopal, and L. T. Pilleggi. Taco: Timing analysis with coupling. In *Proc. DAC*, pages 266–269, 2000.

[13] M. Becer, V. Zolotov, R.Panda, A. Grinshpon, I. Algor, R. Levy, and C. Oh. Pessimism reduction in crosstalk noise aware sta. In *Proc. ICCAD*, pages 954–961, 2005.

[14] K. P. Belkhale and A. J. Suess. Timing analysis with known false subgraphs. In *Proc. ICCAD*, pages 736–739, 1995.

[15] D. Blaauw, S. Sirichotiyakul, and C. Oh. Driver modeling and alignment for worst-case delay noise. In *IEEE Trans. on VLSI*, pages 157–166, 2003.

[16] D. S. Boning and S. Nassif. Models of process variations in device and interconnect. In *Design of High Performance Microprocessor Circuits, chapter 6*, page 6. IEEE Press, 2000.

[17] Y. Cao and L. T. Clark. Mapping statistical process variations toward circuit performance variability: An analytical modeling approach. In *IEEE Trans. on CAD*, volume 26, pages 1866–1873, Oct. 2007.

[18] D. Chai, A. Kondratyev, Y. Ran, K. H. Tseng, Y. Watanabe, and M. M. Sadowska. Temporofunctional crosstalk noise analysis. In *Proc. DAC*, pages 860–863, 2003.

[19] H. Chang and S. Sapatnekar. Statistical timing analysis considering spatial correlations using a single pert-like traversal. In *Proc. ICCAD*, pages 621–625, 2003.

[20] L. Chen and M. Sadowska. Aggressor alignment for worst-case coupling noise. In *Proc. ISPD*, pages 48–54, 2000.

[21] L. C. Chen, S. K. Gupta, and M. A. Breur. A new gate delay model for simultaneous switching and its applications. In *Proc. DAC*, pages 289–294, 2001.

[22] P. Chen, D. Kirkpatrick, and K. Keutzer. Miller factor for gate-level coupling delay calculation. In *Proc. ICCAD*, pages 68–73, 1998.

[23] P. Chen, Y. Kukimoto, and K. Keutzer. Refining switching window by time slots for crosstalk noise calculation. In *Proc. ICCAD*, pages 583–586, 2002.

[24] J. Croix and D. Wong. Blade and razor: Cell and interconnect delay analysis. In *Proc. DAC*, pages 386–389, 2003.

[25] F. Dartu, N. Menezes, and L. T. Pileggi. Performance computation for precharacterized cmos gates with rc loads. In *IEEE Trans. on CAD*, pages 544–553, 1996.

[26] F. Dartu, N. Menezes, J. Qian, and L. T. Pillage. A gate-delay model for high speed cmos circuits. In *Proc. DAC*, pages 576–580, 1994.

[27] D. Das, K. Killpack, C. Kashyap, A. Jas, and H. Zhou. Pessimism reduction in coupling-aware static timing analysis using timing and logic filtering. In *Proc. ASP-DAC*, pages 486–491, 2008.

[28] L. Ding, P. Tehrani, and A. Kasnavi. Determining equivalent waveforms for distorted waveforms. In *U.S. Patent 7272807*, 2007.

[29] T. Fukuoka, A. Tsuchiya, and H. Onodera. Worst-case delay analysis considering the variability of transistors and interconnects. In *Proc. ISPD*, pages 35–42, 2007.

[30] R. Gandikota, D. Blaauw, and D. Sylvester. Interconnect performance corners considering crosstalk noise. In *to appear in Proc. ICCD*, 2009.

[31] R. Gandikota, K. Chopra, M. Becer, D. Blaauw, and D. Sylvester. Top-k aggressors sets in delay noise analysis. In *Proc. DAC*, pages 174–179, 2007.

[32] R. Gandikota, K. Chopra, D. Blaauw, and D. Sylvester. Modeling crosstalk in statistical static timing analysis. In *Proc. DAC*, pages 974–979, 2008.

[33] R. Gandikota, K. Chopra, D. Blaauw, and D. Sylvester. Victim-alignment in crosstalk-aware timing analysis. In *submission to the IEEE Trans. on CAD*, 2008.

[34] R. Gandikota, K. Chopra, D. Blaauw, D. Sylvester, M. Becer, and J. Geada. Victim-alignment in crosstalk-aware timing analysis. In *Proc. ICCAD*, pages 698–704, 2007.

[35] R. Gandikota, L. Ding, P. Tehrani, and D. Blaauw. Worst-case aggressor-victim alignment with current-source driver models. In *Proc. DAC*, 2009.

[36] A. Ghosh, S. Devadas, K. Keutzer, and J. White. Estimation of average switching activity in combinational and sequential circuits. In *Proc. DAC*, pages 253–259, 1992.

[37] P. Gross, R. Arunachalam, K. Rajagopal, and L. Pileggi. Determination of worst-case aggressor alignment for delay calculation. In *Proc. ICCAD*, pages 212–219, 1998.

[38] S. K. Gupta, L. C. Chen, and M. A. Breur. A new gate delay model for simultaneous switching and its applications. In *Proc. DAC*, pages 289–294, 2001.

[39] M. Hashimoto, Y. Yamada, and H. Onodera. Equivalent waveform propagation for static timing analysis. In *IEEE Trans. on CAD*, pages 498– 508, 2004.

[40] S. H.Choi, F. Dartu, and K. Roy. Timed pattern generation for noise-on-delay calculation. In *Proc. DAC*, pages 870–873, 2002.

[41] W. Hoeffding. Probability inequalities for sums of bounded random variables. In *Journal of the American Statistical Association*, volume 58, pages 13–30, 1963.

[42] F. Huebbers, A. Dasdan, and Y. Ismail. Computation of accurate interconnect process parameter values for performance corners under process variations. In *Proc. DAC*, pages 797–800, 2006.

[43] F. Huebbers, A. Dasdan, and Y. Ismail. Multi-layer interconnect performance corners for variation-aware timing analysis. In *Proc. ICCAD*, pages 713–718, 2007.

[44] H. Im, M. Song, T. Hiramoto, and T. Sakurai. Physical insight into fractional power dependence of saturation current on gate voltage in advanced short-channel mosfets (alpha-power law model). In *Proc. ISLPED*, pages 13–18, 2002.

[45] A. B. Kahng, B. Liu, and X. Xu. Statistical gate delay calculation with crosstalk alignment considered. In *Proc. GLSVLSI*, pages 223–228, 2006.

[46] A. B. Kahng, S. Muddu, and E. Sarto. On switch factor based analysis of coupled rc interconnects. In *Proc. DAC*, pages 79–84, 2000.

[47] I. Keller, K. Tseng, and N.Verghese. A robust cell-level crosstalk delay change analysis. In *Proc. ICCAD*, pages 147–154, 2004.

[48] J. Le, X. Li, and L. T. Pileggi. Stac: Statistical timing analysis with correlation. In *Proc. DAC*, pages 343–348, 2004.

[49] R. Levy, D. Blaauw, G. Braca, A. Dasgupta, A. Griashpon, C. Oh, B. Orshay, S. Sirichotiyakul, and V. Zolotov. Clarinet: A noise analysis tool for deep submicron design. In *Proc. DAC*, pages 233–238, 2000.

[50] Y. Liu, S. R. Nassif, L. Pileggi, and A. J. Strojwas. Impact of interconnect variations on the clock skew of a gigahertz microprocessor. In *Proc. DAC*, pages 168–171, 2000.

[51] N. Lu. Statistical and corner modeling of interconnect resistance and capacitance. In *Proc. CICC*, pages 853–856, 2006.

[52] J. D. Ma and R. A. Rutenbar. Interval-valued reduced-order statistical interconnect modeling. volume 26, pages 1602–1613, 2007.

[53] M. Orshansky and K. Keutzer. A general probabilistic framework for worst-case timing analysis. In *Proc. DAC*, pages 556–561, 2002.

[54] A. Papoulis and S. U. Pillai. Probability, random variables and stochastic processes. In *ISBN 0073660116, 4th edition*, 2002.

[55] J. Qian, S. Pullela, and L. T. Pillage. Modeling the effective capacitance of rc interconnect. In *IEEE Trans. on CAD*, pages 1526–1535, 1994.

[56] V. Raghavan and R. Rohrer. A new nonlinear driver model for interconnect analysis. In *Proc. DAC*, pages 561–566, 1991.

[57] V. Rajappan and S. Sapatnekar. An efficient algorithm for calculating the worst-case delay due to crosstalk. In *Proc. ICCD*, pages 76–81, 2003.

[58] Y. Ran, A. Kondratyev, K. H. Tseng, Y. Watanabe, and M. M.Sadowska. Eliminating false positives in crosstalk noise analysis. In *IEEE Trans. on CAD*, pages 1406–1419, 2005.

[59] J. Rubinstein, P. Penfield, and M. A. Horowitz. Signal delay in rc tree networks. In *IEEE Trans. on CAD*, volume 2, pages 202–211, July 1983.

[60] S. Sapatnekar. Timing. In *ISBN 978-1-4020-7671-8*, 2004.

[61] S. S. Sapatnekar. A timing model incorporating the effect of crosstalk on delay and its application to optimal channel routing. *IEEE Trans. on CAD*, pages 550–559, 2000.

[62] Y. Sasaki and G. D. Micheli. Crosstalk delay analysis using relative window method. In *Proc. IEEE ASIC/SOC Conference*, pages 9–13, 1999.

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

[64] T. Sato, Y. Cao, D. Sylvester, and C. Hu. Characterization of interconnect coupling noise using in-situ delay change curve measurements. In *Proc. IEEE ASIC/SOC Conference*, pages 321–325, 2000.

[65] K. L. Shepard and V. Narayanan. Noise in deep submicron digital design. In *Proc. ICCAD*, pages 524–531, 1996.

[66] K. L. Shepard, V. Narayanan, P. C. Elmendorf, and G. Zheng. Globalharmony: Coupled noise analysis for full-chip rc interconnect networks. In *Proc. ICCAD*, page 139146, 1997.

[67] D. Sinha, G. Schaeffer, S. Abbaspour, A. Rubin, and F. Borkam. Constrained aggressor set selection for maximum coupling noise. In *Proc. ICCAD*, pages 790–796, 2008.

[68] D. Sinha and H. Zhou. A unified framework for statistical timing analysis with coupling and multiple input switching. In *Proc. ICCAD*, pages 837–843, 2005.

[69] D. Sinha and H. Zhou. Statistical timing analysis with coupling. In *IEEE Trans. on CAD*, pages 524–531, 2006.

[70] S. Sirichotiyakul, D. Blaauw, C. Oh, R. Levy, V. Zolotov, and J. Zuo. Driver modeling and alignment for worst-case delay-noise. In *Proc. DAC*, pages 720–725, 2001.

[71] R. Tayade, V. K. Kalyanam, S. Nassif, M. Orshansky, and J. Abraham. Estimating path delay distribution considering coupling noise. In *Proc. GLSVLSI*, pages 61–66, 2007.

[72] C. Visweswariah, K. Ravindran, K. Kalafala, S. G. Walker, and S. Narayan. First-order incremental block-based statistical timing analysis. In *Proc. DAC*, pages 331–336, 2004.

[73] J. M. Wang, O. Hafiz, and P. Chen. A non-iterative model for switching window computation with crosstalk noise. In *Proc. ASP-DAC*, pages 846–851, 2004.

[74] S. C. Wong, G. Y. Lee, and D. J. Ma. Modeling of interconnect capacitance, delay, and crosstalk in vlsi. In *IEEE Trans. on Semiconductor Manufacturing*, volume 13, pages 108–111, 2000.

[75] T. Xiao and M. Sadowska. Worst delay estimation in crosstalk aware static timing analysis. In *Proc. ICCD*, pages 115–120, 2000.

[76] W. Zhao and Y. Cao. Predictive technology model for nano-cmos design exploration. In *J. Emerg. Technol. Comput. Syst.*, volume 3, pages 1550–4832, 2007.

[77] H. Zhou, N. Shenoy, and W. Nicholls. Timing analysis with crosstalk is a fixpoint on a complete lattice. In *Proc. DAC*, pages 714–719, 2001.