

Towards True Crosstalk Noise Analysis

Pinhong Chen Kurt Keutzer
Dept. of EECS, Univ. of California at Berkeley
Berkeley, CA94720, USA

Abstract

Accurate noise analysis is currently of significant concern to high-performance designs, and the number of signals susceptible to noise effects will certainly increase in smaller process geometries. Our approach uses a combination of temporal and functional information to eliminate false transition combinations and thereby overcome insufficiencies in static noise analysis. A similar idea arises in timing analysis where functional and timing information is used to eliminate false paths. The goal of our work is to develop an algorithm, software tool, and noise analysis flow that provide an accurate and conservative approach to noise analysis. In particular, this paper proposes an approach to identifying a pair of vectors that exercises the maximum crosstalk noise.

1 Introduction

Accurate noise analysis is of increasing and significant concern to designs containing dynamic logic or to designs which have on-chip analog portions. It is expected that the impact of noise effects will increase in smaller process geometries. The current approaches to interconnect crosstalk analysis are usually based on identifying the spatial relationship between two coupling signals and then adding a *static analysis* of the temporal relationship [1, 2]. The use of static timing information in this methodology is similar to the static timing analysis without false-path elimination, so it may lead to an overly pessimistic estimation of the actual noise in the circuit. When static noise analysis information is inaccurate it may result in the expenditure of additional effort in the form of unnecessarily re-routing of wires or modifying signal drivers. An even greater drawback of static analysis approach is that it may fail to correctly analyze signal glitches which can also be responsible for erroneous switching in the circuit.

The key to improving the accuracy in static timing analysis was to add functional information and thereby compute the “true delay” of the circuit. Similarly, we believe the key to improving noise analysis is to also to add functional information to the temporal. By analogy, we call this computing the “true crosstalk noise” of the circuit.

In this paper, we present a method for searching for the vector pair which maximizes the crosstalk noise for a combinational sub-

circuit. This search uses the timing information for the relevant signals together with the functional information of the gates computing the signals. Putting these two together a tighter upper bound can be obtained and the input vectors that exercise the maximum crosstalk noise can be identified.

In the following sections, we first review some useful methods and related work for crosstalk analysis in Section 2. Due to the huge vector search space, some techniques to prune this search space are introduced in Section 3. Also, we will discuss several models with respect to their complexity, efficiency, and accuracy for computing the crosstalk noise bound. The vector search algorithm for the maximum crosstalk noise is explained in Section 4. Section 5 shows the experimental results. Some areas for improvement are discussed in Section 6.

2 Approaches and Related Work

The most straightforward approach to finding a vector pair that maximizes the crosstalk noise is to simply exhaustively simulate all input vector pairs. This is rarely computationally feasible. Therefore, an accurate and computationally efficient method needs to find the vector pair that stimulates the maximum crosstalk noise. Instead of exhaustive logic simulation, we formulate this problem as the Boolean Constraint Optimization Problem (BCOP) to solve it exactly. Moreover, our approach can be extended to a general vector-search scheme, which can be used to search for the vectors causing the maximum IR drop or the maximum power consumption [3].

We will begin by reviewing prior work that has some elements in common with the the problem of maximum crosstalk noise analysis. In [4], the authors propose a multiple-value logic to generate an input vector to test cross coupling faults. However, there is no timing involved, and thus it is unable to find the real noise bound, but just a pair of vectors for circuit testing.

In [3], the authors provide an approach to finding a vector pair which maximizes power dissipation. This problem shares a couple of elements in common with our own: function and timing must be integrated and failure to analyze glitches will lead to a non-conservative procedure. Unfortunately, the search approach shown in [3] is somewhat primitive and is not able to scale to the size of problem we wish to consider.

The use of Timed Boolean Functions (TBF) [5] has also been proposed for computation of the vector pair causing the maximum number of transitions. This is related to our problem but not equivalent, and no strategy for noise analysis is reported there. The underlying computational mechanism is based on Binary Decision Diagrams (BDDs) [6]. Also basing his approach on BDDs is [2]. This work does focus on crosstalk noise avoidance and does a nice

job of describing the relationship between spatial, temporal, and functional elements of noise analysis; however, in [2] results are cited on only very modestly sized circuits (< 1000 gates). BDD-based techniques are necessarily limited because many circuits do not have compact BDD representations.

Other recent work has used Timed Automata to model coupling delays [7]; however, these approaches are even more computationally expensive than the BDD-based approaches.

Another relevant reference is [8], where the authors show that using Miller capacitance $2C_c$, twice the coupling capacitance, to bound the worst case delay may not be correct.

In conclusion, while a variety of work has been performed that is relevant to this problem there is still a need for an accurate “true noise” analysis approach that is computationally efficient. This is the goal of this work.

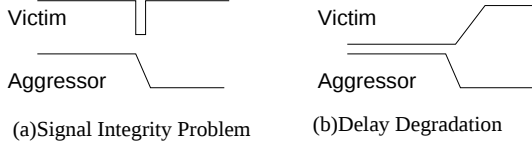


Figure 1: Signal Integrity and Delay Degradation

2.1 Varieties of Noise Effects

In general, crosstalk noise can affect the circuit functionality in two ways: one is signal integrity and the other is delay degradation[1, 2]. The former case happens when the noise couples to a static signal, causing an erroneous logic voltage level such as shown in Fig. 1(a). For precharge logic, this is a deleterious effect. The latter case happens when the noise couples to a switching signal, causing the delay degradation such as shown in Fig. 1(b). In this paper, we are concerned with the former case, while the latter case can be pursued by careful design of the objective function.

3 Search Space Pruning

In practice, the space for the crosstalk noise is too huge for searching. The spatial, temporal, and functional properties of a circuit[2] closely interact to result in the significant crosstalk noise. These properties are described in the following sections.

Since the only concern is the significant noise which is greater than an allowable margin, these properties can then be used to reduce the search space by excluding some of the conditions which apparently do not cause any significant noise on a victim net.

3.1 Spatial Pruning

Two adjacent nets in a layout may be susceptible to crosstalk. Using a simplified lumped RC model, the noise level can be calculated as Eq. 1[4]. Considering a pair of victim and aggressor nets in the physical layout, the waveform on the victim net can be computed as[4]:

$$V_v(t) = \begin{cases} V_{DD} - V_{DD} \frac{C_x R_v}{t_f} (1 - e^{-t/R_v C_t}), & 0 < t \leq t_f \\ V_v(t_f) e^{-(t-t_f)/R_v C_t} + V_{DD} (1 - e^{-(t-t_f)/R_v C_t}), & t > t_f \end{cases} \quad (1)$$

where $C_t = C_v + C_x$. Please see Fig. 2, where C_v, C_a, C_x are the capacitance of victim net, aggressor net, and coupling capacitance, respectively; R_v is the resistance of victim net. If multiple aggressor nets have the coupling effect on the victim net, C_t should be modified as the sum of C_v and all the coupling capacitances. The

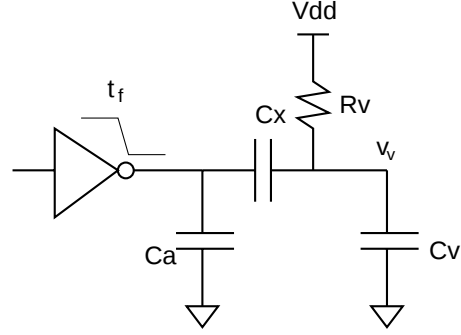


Figure 2: Noise Level Model

total coupling effect can be calculated as a superposition of each coupling effect using Eq. 1, that is,

$$\Delta V_v(t) = \sum_i \begin{cases} 0, & 0 < t \leq t_i \\ V_{DD} \frac{C_{xi} R_v}{t_{fi}} (1 - e^{-(t-t_i)/R_v C_{ti}}), & t_i < t \leq t_i + t_{fi} \\ V_{DD} \frac{C_{xi} R_v}{t_{fi}} (1 - e^{-t_{fi}/R_v C_{ti}}) e^{-(t-t_i-t_{fi})/R_v C_{ti}}, & t > t_i + t_{fi} \end{cases} \quad (2)$$

where C_{xi} is each coupling capacitance to the aggressor net i , t_i is the time when the aggressor net i begins to fall, and t_{fi} is the fall time of the aggressor net i . The maximum value of ΔV_v occurs when all of $t_i + t_{fi}$ are equal. This value is used in *simple worst case* model, where one assumes the extreme case may happen regardless of timing edge alignment or functionality.

3.2 Temporal Pruning

The crosstalk noise is also related to the timing window of each net. For example, in Fig. 3, the net V's timing window overlaps the net A1's timing window, which means that net V may be attacked by net A1, while the timing window of net A2 is far apart from that of net V and unlikely to have any crosstalk noise on net V. This ap-

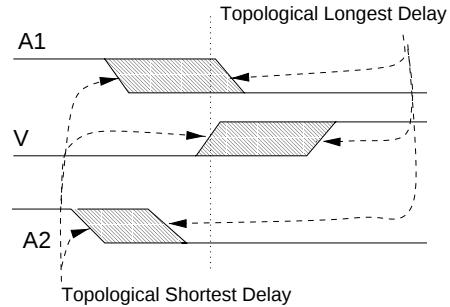


Figure 3: Temporal Relationship between Victim Net and Aggressor Nets

proach is reflected in *static noise analysis* model, where functional information is ignored but timing information is included. In this model, if an aggressor signal shares a valid timing window with respect to a victim signal then one makes the conservative assumption that the transitions were correlated so as to create maximum crosstalk.

3.3 Functional Pruning

Applying temporal and spatial pruning alone may also lead to inaccuracies. Just as functional pruning can be used in static timing

analysis to eliminate paths that are never responsible for the delay of the circuit, functional pruning can be used in noise analysis to eliminate those signals/paths that can never be responsible for noise problems because of their functional relationship. By exploring the functional space, we can search for a pair of input vectors whose function allows them to create the maximum noise. This approach is reflected in the *zero-delay model*, where temporal information is ignored but functional information is included. In this model, if an aggressor signal transitions against a victim signal within a clock cycle then one makes the conservative assumption that the transitions were correlated so as to create maximum crosstalk.

3.4 Problem Complexity vs Accuracy

The maximum crosstalk noise depends on the spatial, temporal, and functional aspects of a circuit, so it is generally not easy to be exactly determined. Several techniques have been proposed [9, 8, 2] for bounding the maximum crosstalk noise. In general, the spatial information can be easily considered.

In the *floating mode delay model* [10], the prior state of both the victim and aggressor signals are ignored. In this model, both signals are initially assumed to be in an indeterminate state. If there exists a vector that causes both the victim and aggressor to arrive at opposite values with temporal correlation then the conservative assumption is made that the prior state of each signal (before the single vector was applied) will set up the signals for maximum crosstalk. Finally, the fixed-delay 2-vector model considers both timing and functional completely.

The waveform of each method may look like Fig. 5. In the simple worst case, only spatial pruning is used so that every aggressor is assumed to switch in the opposite direction of the victim net V. The zero-delay model ignores the timing effects, while the static approach ignores the functional effects. The fixed-delay model considers the timing and functional effects at the same time.

The relationship between these techniques can be shown in Fig. 4. The maximum accuracy results from the maximum integration of functional and timing information. Thus the 2-vector approach is more accurate than any other named. The relative accuracy between the other approaches will vary from circuit to circuit.

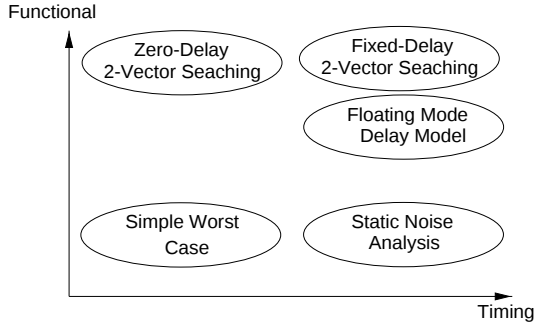


Figure 4: Complexity of Crosstalk Noise Analysis

Even excluding the timing information, such as the zero-delay model, we still get the Boolean Constraint Optimization Problem (BCOP), which is equivalent to the Binate Covering Problem (BCP). The static analysis method can result in a linear programming problem [1]. However, when the functional aspect is involved, it becomes the BCOP.

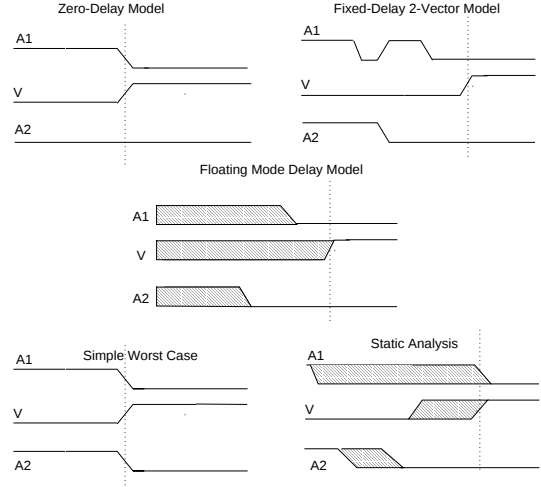


Figure 5: Complexity of Crosstalk Noise Analysis

4 Vector Pair Searching Algorithm

4.1 Overview

Our proposed approach is based on a similar method of the functional timing verification [10, 11, 12]. We need to set up timed Boolean variables for each node at some time points and conjunct the gate characteristic functions, which represent all the possible logic combinations of a gate. According to the allowable noise level to assign the weights of some Boolean variables, the vector pair search problem can be formulated as the BCOP, in which we find some assignment to satisfy the constraint and maximize an objective function in terms of these Boolean variables. In the following, we describe each step in details.

4.2 BCOP: Boolean Constraint Optimization Problem

Given an objective function,

$$h(x_1, x_2, \dots, x_n) = c_1x_1 + c_2x_2 + \dots + c_nx_n,$$

where $x_1, x_2, \dots, x_n \in B = \{0, 1\}$ and $c_1, c_2, \dots, c_n \in R$, we want to maximize $h(x_1, x_2, \dots, x_n)$ such that it satisfies the constraint:

$$f(x_1, x_2, \dots, x_n, \overline{x_1}, \overline{x_2}, \dots, \overline{x_n}) = \prod_i \sum_j x_{ij} \equiv 1,$$

where $x_{ij} \in \{x_1, x_2, \dots, x_n, \overline{x_1}, \overline{x_2}, \dots, \overline{x_n}\}$. The constraint can be written in the Conjunction Normal Form (CNF). This problem is similar to the BCP.

For weighting a switching on the aggressor nets, the cost function may not be linear. Moreover, if timing is considered, some weighted terms must be represented as a complex conjunction of some circuit variables. In such cases, we implement the branch-and-bound algorithm to address this special objective function evaluation.

4.3 Constructing Circuit via SAT

To represent a combinational circuit, we use the characteristic function of each gate and set up a variable for each node. Conjuncting these characteristic functions together, we obtain a characteristic function to represent the whole circuit [13, 14]. These characteristic functions are represented by the CNF clauses. A circuit is functionally consistent if and only if the characteristic function is a tautology or, equivalently, the CNF clause can be satisfied. Namely, we

can find a valid assignment for each variable, which means there is a valid logic combination of the inputs, the outputs and the internal nodes of the circuit.

For example, in Fig.6, this AND circuit can be characterized by

$$f(X, A, B) = (X + \bar{A} + \bar{B})(\bar{X} + A)(\bar{X} + B). \quad (3)$$

f is equal to one if and only if X , A and B are a valid variable

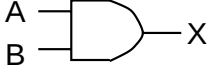


Figure 6: Characteristic Function of an AND Gate

assignment for AND gate operation. Consider the circuit in Fig.7. We can conjunct the characteristic function of each gate together to obtain the characteristic function of this whole circuit.

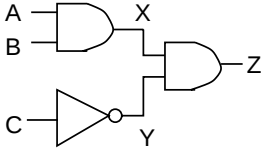


Figure 7: Conjunction of Characteristic Functions

$$\begin{aligned} f(Z, X, Y, A, B, C) = & (X + \bar{A} + \bar{B})(\bar{X} + A)(\bar{X} + B) \\ & \cdot (Y + C)(\bar{Y} + \bar{C}) \\ & \cdot (\bar{Z} + X + Y)(Z + \bar{X})(Z + \bar{Y}). \end{aligned}$$

Thus, the stable state of this circuit must satisfy the constraint above (f equals to 1).

4.4 Maximum Noise under the Zero-Delay Model

In the zero-delay model, all the gates and the interconnect are assumed to have zero delay. Therefore, the maximum noise occurs when all the aggressor nets make transitions in the opposite direction to the victim net. We can then investigate the correlation between signals to find the maximum number of the opposite transitions, which maximize the crosstalk noise under the zero-delay model.

For a single victim net, given the coupling capacitance between each aggressor net and the victim net, we can obtain the maximum noise by

1. Setting up two variables for each node: one variable denotes the value of time 0, and the other is for time ∞ .
2. Build the characteristic function into the CNF clause for two sets of variables.
3. Find the maximum weighted sum of each aggressor's variable, where the weighting is proportional to the coupling capacitance. The phases are assigned according to the switching direction on the aggressor nets.
4. Set up the initial condition that makes the victim net static, that is, the victim net stays on the same logic state even after the second vector applied.

4.5 Fixed Delay Circuit Construction via SAT

The zero-delay model, in general, is not satisfied for an accurate noise bound, and it often gives almost the same scale of the noise bound as the static noise analysis. Thus, timing information is important to be considered in this vector search procedure to obtain a tighter maximum noise bound.

Timing information can be included by introducing timed Boolean variables, with which we can represent the logic values of a node at different time points.

Also, it is assumed that there are 2 input vectors to apply to the primary inputs of a circuit: one vector at time minus infinity, and the other at time zero. The former vector drives each node of the circuit into a known state no matter how long the circuit delay is and the latter one exercises the maximum crosstalk noise. Actually, this scheme can be easily generalized for multiple vectors or extended to the other applications.

4.5.1 Using Timed Boolean Variables

The timed Boolean variable, which is used to represent a logic state for a node at some time point, is a mapping $v : R \rightarrow \{0, 1\}$. The negative timed Boolean variable denoted by $\bar{v}$ is the complemented variable of v .

Suppose the circuit in Fig. 8, where $Z(t = 4)$ is evaluated. $Z(t =$

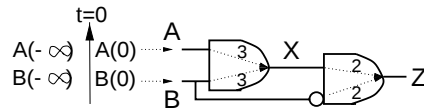


Figure 8: Timed Boolean Variable Example

4) is abbreviated as $Z(4)$. Since the delays between X to Z and B to Z are 2, we can represent the gate on the right by the characteristic function:

$$(Z(4) + \bar{X}(2) + B(2))(\bar{Z}(4) + X(2))(\bar{Z}(4) + \bar{B}(2))$$

Since B is a primary input, only two values are applied. $B(2)$ is equal to $B(0)$. We rewrite the above characteristic function as:

$$(Z(4) + \bar{X}(2) + B(0))(\bar{Z}(4) + X(2))(\bar{Z}(4) + \bar{B}(0)). \quad (4)$$

Similarly,

$$\begin{aligned} & (X(2) + \bar{A}(-1) + \bar{B}(-1)) \\ & (\bar{X}(2) + A(-1))(\bar{X}(2) + B(-1)) \end{aligned}$$

A and B are primary inputs, $A(-1) = A(-\infty)$ and $B(-1) = B(-\infty)$. Therefore,

$$\begin{aligned} & (X(2) + \bar{A}(-\infty) + \bar{B}(-\infty)) \\ & (\bar{X}(2) + A(-\infty))(\bar{X}(2) + B(-\infty)). \end{aligned} \quad (5)$$

The whole circuit characteristic function for $Z(t = 4)$ becomes

$$\begin{aligned} & (Z(4) + \bar{X}(2) + B(0)) \\ & (\bar{Z}(4) + X(2))(\bar{Z}(4) + \bar{B}(0)) \\ & (X(2) + \bar{A}(-\infty) + \bar{B}(-\infty)) \\ & (\bar{X}(2) + A(-\infty))(\bar{X}(2) + B(-\infty)). \end{aligned} \quad (6)$$

Thus, $Z(4) = 1$ can be sensitized by assigning $A(-\infty) = 1$, $B(-\infty) = 1$, $B(0) = 0$, $X(2) = 1$. This can be obtained by solving SAT of the above characteristic function, Eq. 6. The entire waveform is actually as shown in Fig. 9.

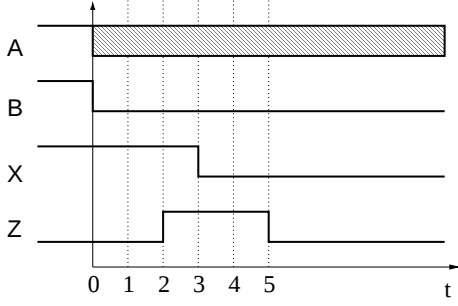


Figure 9: Waveform of Example Circuit in Fig. 9

4.5.2 Boolean Constraint Optimization Problem

The Boolean constraint optimization problem is equivalent to the BCP by transforming the objective function into a cost function, and then minimizing the cost function. The straightforward method to solve this problem is the branch-and-bound method, while many BCP techniques have been proposed [15].

Some heuristics may be possible to reduce the complexity such as a coarse quantum time, relaxing SAT formulation, or partial variable collapsing. These techniques will be discussed in a later paper.

4.5.3 Discrete Required Time Analysis

In general, the required times under the fixed delay model are not continuous, that is, the possible combinational path delays are finite, resulting in discrete path delays and hence the discrete required times. Therefore, we can analyze these possible discrete required times to reduce the number of timed Boolean variables. This could be done by the topological sorting algorithm.

4.5.4 Structural Hashing

In order to reduce the number of timed Boolean variables, this technique tries to find all possible reuses of the timed Boolean variable from the circuit network structure. The simplest reuse is shown in Fig. 10, where we can reuse $\bar{A}$ for X and replace any occurrence of X by $\bar{A}$.

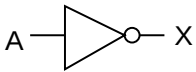


Figure 10: Variable Reuse

When each multi-input gate is represented by or decomposed into a cube, the localized normal form for gate function representation is established. A hash table can be used to store the output variables with the sorted input list of each gate as a hashing key. The reuse of the circuit in Fig. 11 can be found by the first reuse $C = \bar{A}$ and then $Y = \bar{X}$. This technique can reduce the number of

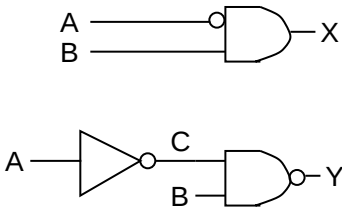


Figure 11: Variable Reuse

Circuit	Fixed Delay	Static Noise Analysis	Zero Delay	Simple Worst
C0432	1.72V	1.83V	1.94V	1.94V
C0499	0.84V	1.50V	1.61V	2.15V
C0880	1.65V	1.65V	2.04V	2.04V
C1355	0.78V	0.78V	1.66V	2.21V
C1908	1.40V	1.61V	1.07V	2.15V
C2670	1.19V	1.61V	1.07V	2.15V
C3540	0.43V	0.43V	1.72V	1.72V
C5315	0.54V	0.54V	2.15V	2.15V
C7552	0.27V	0.29V	1.13V	1.50V

Table 1: Comparison of Maximum Noise Bound(Maximum ΔV_v)

variables and clauses dramatically, and even reduce the redundant variables at different time points.

5 Experimental Results

Unfortunately, those circuits for which we had functional information we did not have layout information, and those circuits for which we had layouts we did not have functional information. To test our approach we used the ISCAS85 benchmark circuit set and made some simple assumptions that would emulate accurate layout information.

In actual practice, due to the locality of layout (i.e. spatial pruning) for each victim net, there are typically only a few aggressors which can cause significant noise on it. For testing our approach, we emulate this effect by simply choosing four random aggressor nets and one victim net. Our results are shown in Table 1. It may be useful to refer back to Section 3.4 to understand the approaches associated with the columns. The first column gives the circuit name from the ISCAS benchmark. The column entitled *simple worst* is just the sum of the maximum crosstalk noise from each aggressor. The columns entitled *the zero-delay model* and *static noise analysis* are as described in Section 3.2 and 3.3, respectively. The column entitled *fixed delay* is the 2-vector approach described in Section 3.4.

Since we don't have real layout information, we emulate some electrical parameters, such as C_v, C_x, R_v, t_f , based on a sample of 0.5um 5V static CMOS process and use Eq. 2 to calculate the maximum ΔV_v , the maximum voltage difference shown on the victim net. Arbitrarily, the same electrical parameters are used for each circuit. We assume the condition similar to Fig. 2, in which the four aggressor nets possibly make a transition from high to low, and the victim net keeps static high. The numbers shown in Table. 1 are the maximum voltage difference on the victim net due to the coupling effect.

For the minimum voltage to be regarded as logic high, it should be within 70% of VDD, that is, ΔV_v should be less than 1.5 Volts in our test case. Therefore, the comparison of the maximum noise bound shows that for C499, C1908 and C2670 circuits, the static noise analysis makes an over-pessimistic prediction while the fixed delay model does not. The zero-delay model can not take into account the effect of glitches, and as a result it under-predicts the maximum noise bound of C1908 and C2670. For most of the cases, the zero-delay model can not predict tighter maximum noise bound than the static noise analysis. The simple worst case gives little information and is included only as a reference. C6288 could not be completed within reasonable time.

6 Future Work

Our approach aims to find two vectors that maximize the noise under assumptions that are as accurate as possible while being conservative. One way to improve the conservatism of our approach is to consider the effect of cross-coupling on delay degradation, and therefore on timing. It is possible to model the rise waveform on the victim net and compute for the maximum delay degradation using Eq. 2. However, as more accurate RC-interconnect model is desired for deep submicron technology, the modeling approach similar to [8, 9] should be taken into account.

One area for improvement in the accuracy of our approach is to consider combinational logic blocks in their sequential context. We consider combinational blocks in isolation and presume that the vector pair that we identify is always within the valid sequential state-space of the circuit. In other words we assume that the vector pair that we identify can actually be excited in the normal operation of the circuit. This may not be true and thus we may over estimate the noise of the circuit if this is not so. Resolving this issue is more computationally challenging as it is equivalent to the sequential testing problem, or alternatively, the sequential state space reachability problem, which is currently unsolved.

One approach to improving both the accuracy and the conservatism of our method is to incorporate a timing model in which bounded delay intervals, rather than fixed delay values, are used. This approach will be investigated but currently it appears to make the problem computationally intractable for reasonable sized circuits.

As we were unable to complete our computation on C6288 there is still room for improvement in improving the computational performance of our approach.

7 Conclusions

The goal of our work is to develop an algorithm, software tool, and noise analysis flow that provide an accurate and conservative approach to analysis of noise problems that could cause voltage glitches leading to erroneous switching of dynamic logic or the malfunctioning of analog circuitry. With such a tool available, the time consuming manual work in analyzing potentially noisy signals could be avoided. To achieve maximum accuracy our approach finds two vectors that maximize the noise, and we have presented a general scheme for identifying the proper vector pair.

This paper compares the results obtained by simpler methods including the *zero delay model* in which functional information is incorporated but timing information is neglected and the *static noise analysis* approach in which temporal information is incorporated by functional information is ignored. Our approach is shown to be strictly more accurate than either of these approaches, while still being computational feasible on industrially sized sub-circuits.

References

- [1] K. L. Shepard. "Design Methodologies for Noise in Digital Integrated Circuits". In *Design Automation Conference*, pages 94–99, 1998.
- [2] D. A. Kirkpatrick. "*The Implication of Deep Sub-micron Technology on the Design of High Performance Digital VLSI System*". PhD thesis, CAD Group Ph.D. Dissertation, U.C. Berkeley, 1997.
- [3] S. Devadas, K. Keutzer, and J. White. "Estimation of Power Dissipation in CMOS Combinational Circuit Using Boolean Function Manipulation". *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 11:373–383, Mar. 1992.
- [4] A. Rubio, N. Itazaki, X. Zu, and K. Kinoshita. "An Approach to the Analysis and Detection of Crosstalk Faults in Digital VLSI Circuits". *IEEE Trans. on Computer-Aided Design*, 13:387–394, Mar. 1994.
- [5] W. K.C. Lam and R. K. Brayton. "*Timed Boolean Functions*". Kluwer Academic Publishers, 1994.
- [6] R. E. Bryant. "Graph-based Algorithms for Boolean Function Manipulation". *IEEE Trans. on Computers*, pages 677–691, Aug. 1986.
- [7] S. Tasiran et. al. "Comuting Delay with Coupling with Timed Automata". In *Tau 97 : International Workshop ion Timing Issues*, pages 232–242, 1997.
- [8] G. Yee, R. Chandra, V. Ganesan, and C. Sechen. "Wire Delay in the Presence of Crosstalk". In *IEEE/ACM International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems*, pages 170–175, 1997.
- [9] K. L. Shepard, V. Narayanan, P.C. Elmendor, and Gutuan Zheng. "Global Harmony: Coupled Noise Analysis for Full-Chip RC Interconnect Network". In *International Conference on Computer Aided Design*, pages 139–146, 1997.
- [10] S. Devadas, K. Keutzer, and S. Malik. "Computation of floating mode delay in combinational networks: Theory and algorithms". *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 12:1913–1923, Dec. 1993.
- [11] S. Devadas, K. Keutzer, S. Malik, and A. Wang. "Certified Timing Verification and the Transition Delay of a Logic Circuit". *IEEE Trans. on Very Large Scale Integration Systems*, 2:333–342, Sep. 1994.
- [12] T. Sasao(ed). "*Logic Synthesis and Optimization, Ch.8: Delay Models and Exact Timing Analysis*". Kluwer Academic Publishers, 1993.
- [13] T. Larrabee. "Test Pattern Generation Using Boolean Satisfiability". *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 11:4–15, Jan. 1992.
- [14] P. Stephan, R. K. Brayton, and A. L. Sangiovanni-Vicentelli. "Combinational Test Generation Using Satisfiability". *IEEE Trans. on Computer-Aided Design*, 19:4–15, Sep. 1996.
- [15] O. Coudert. "On Solving Covering Problem". In *Design Automation Conference*, pages 197–202, 1996.