Refining Switching Window by Time Slots for Crosstalk Noise Calculation

Pinhong Chen*+ Yuji Kukimoto* Kurt Keutzer+
+Dept. of EECS, U.C. Berkeley
Berkeley, CA94720

*Cadence Design System Inc.
555 River Oaks Parkway
San Jose, CA 95134

ABSTRACT

For crosstalk noise calculation, computing switching windows of a net helps us identify noise sources accurately. Traditional approaches use a single continuous switching window for a net. Under this model, signal switching is assumed to happen any time within the window. Although conservative and sound, this model can result in too much pessimism since in reality the exact timing of signal switching is determined by a path delay up to the net, i.e. the underlying circuit structure does not always allow signal switching at arbitrary time within the continuous switching window. To address this inherent inaccuracy of the continuous switching window, we propose a refinement of the traditional approaches, where signal switching is characterized by a set of discontinuous switching windows instead of a single continuous window. Each continuous switching window is divided into multiple windows, called time slots, and the signal switching activity of each slot is analyzed separately to calculate the maximum noise with more accuracy. By controlling the size of a time slot we can trade off accuracy and runtime, which makes this approach highly scalable. We have confirmed by experiments on industrial circuits that up to 90% of the noise violations detected by the traditional approach can be unreal.

1. INTRODUCTION

As the process technologies advance, coupling capacitances become dominant for the total capacitance due to the change of the aspect ratio of metal wires in deep sub-micron designs. The impact of this trend is that coupling noise affects gate/net delays more significantly, resulting in larger timing variation and lower yield rate of a chip. Many approaches have been proposed to take these effects into account in static timing analysis(STA)[1, 2, 3, 4, 5, 6, 7, 8].

Crosstalk noise affects timing in two ways. If two nets switch at about the same time in the same direction, the delay decreases, and for the opposite directions, the delay increases. If the adjacent nets are quiet, there is no crosstalk noise. Therefore, it is important to identify the **switching window**(or timing window), a timing interval during which a net or a timing node can possibly make transitions, so that if there is no overlap of switching windows between two coupling nodes, we can immediately conclude that there is no timing variation, thereby reducing the analysis pessimism. Finding the maximum crosstalk noise is thus equivalent to finding a maximum weighted sum of channel density problem[9].

Typically, the switching windows considered in the literature[1, 2, 3, 4, 5, 6, 7, 8] are continuous(see Figure 1). They are a timing interval from the earliest arrival time to the latest arrival time of a net. However, the arrival times are typically not continuous inside a switching window(see Figure 2), since there are only a finite number of paths to a net. Consider Figure 1 and 2. Figure 2 captures switching activity more accurately by discontinuous windows, while Figure 1 is an approximation of Figure 2 by a continuous window. Suppose net *A* and net *B* are aggressors to be aligned for the maximum noise. As is clear from Figure 2, there is no switching window overlap. If Figure 2 is approximated by Figure 1, however, the two windows have overlap, resulting in a false alignment and a pessimistic noise estimation. The goal of this paper is to take advantage of discontinuous switching windows to calculate crosstalk noise more accurately.

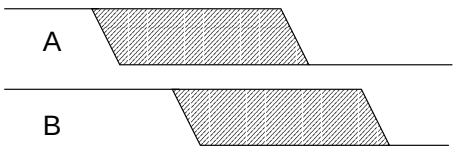

Figure 1: Continuous Switching Windows

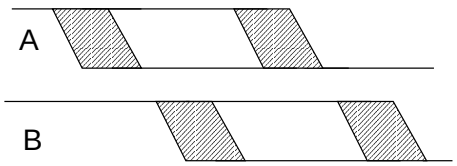

Figure 2: Discontinuous Switching Windows

Using a fixed delay model, the maximum number of discrete arrival times at a net is equal to the number of paths to the net. To avoid

0-7803-7607-2/02/$17.00

handling a potentially exponential number of discrete arrival times, we use a time slot approach, where a continuous switching window is refined as a set of time slots of the same size. The size of a time slot is thus an effective scaling factor to trade off analysis accuracy versus speed and capacity. As the analysis resolution goes finer, more maximum noise can be justified precisely.

The rest of this paper is organized as follows. In Section 2. we introduce the formulation and algorithmic aspects of our approach, and discuss the theory behind it. Section 3 addresses the resolution issues. Experimental results are shown in Section 4. In Section 5, we investigate a refinement of proposed method, where slew effects on the maximum noise are modeled more accurately. Our conclusion is given in Section 6.

2. FORMULATION AND ALGORITHM

A **victim** is a net that suffers from a noise effect, and an **aggressor** is a net that contributes noise. The roles may change depending on the context. An **index set** denotes a collection of net indices, representing a subset of all nets. Let d_{ki} be the interconnect delay of net k targeting net i, A_i be an index set of aggressor nets of net i, D_i be an index set of fanin nets of net i's driving gate, and δ_{ki} be the gate delay from a fanout end of net k to net i. Let x_i and y_i represent

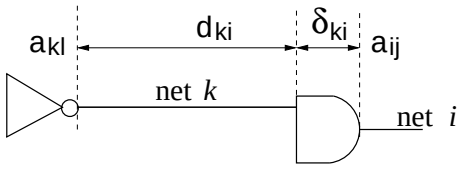

Figure 3:

the latest and the earliest arrival times of net i, respectively. They can be written as:

$$x_i = \max_{k \in D_i}(x_k + d_{ki} + \delta_{ki}) \tag{1}$$

and

$$y_i = \min_{k \in D_i}(y_k + d_{ki} + \delta_{ki}), \tag{2}$$

respectively. Traditional continuous switching window model defines the switching window from time y_i to time x_i.

Let s be the size of the time slot and $a_{ij} \in \{0,1\}$ be a Boolean variable defined as follows:

$$\begin{cases} a_{ij} = 1, & \text{if net } i \text{ has any arrival time } t \\ & \text{such that } js \le t < (j+1)s. \\ a_{ij} = 0, & \text{otherwise.} \end{cases}$$

It can be recursively computed as:

$$a_{ij} = \bigvee_{k \in D_i} a_{kl}, \tag{3}$$

where $\bigvee$ is a logical "OR" operator and $l = j - \lfloor \frac{d_{ki}+\delta_{ki}}{s} \rfloor$. Strictly speaking, we may need to check two time slots:

$$\lfloor j - \frac{d_{ki}+\delta_{ki}}{s} \rfloor \le l \le \lfloor j + 1 - \frac{d_{ki}+\delta_{ki}}{s} \rfloor.$$

The nets are visited in the same order as in the continuous switching window calculation or longest path calculation: a depth-first traversal of the direct acyclic graph(DAG). Assuming that aggressors in the same time slot can align to create the maximum noise on the victim net i, we can calculate the maximum noise at net i as:

$$\max_{0 \le j \le \lfloor \frac{L}{s} \rfloor} \sum_{k \in A_i} \phi_{ki} a_{kj}, \tag{4}$$

where L is the longest path delay or the maximum arrival time of the circuit and $\phi_{ki} \in R$ is the noise effect from the aggressor net k to the victim net i. Although the slew rate of net k may not be the same for all the time slots, ϕ_{ki} is calculated using the fastest slew available on net k to ensure a pessimistic analysis. We can further reduce the search range down to the interval $[\lfloor \frac{y^{\min}_{A_i}}{s} \rfloor, \lfloor \frac{x^{\max}_{A_i}}{s} \rfloor]$, where $y^{\min}_{A_i}$ is the minimum arrival time of all the aggressors of net i, and $x^{\max}_{A_i}$ is the maximum arrival time of all the aggressors of net i. That is to say, Eq. 4 becomes:

$$\max_{\lfloor \frac{y^{\min}_{A_i}}{s} \rfloor \le j \le \lfloor \frac{x^{\max}_{A_i}}{s} \rfloor} \sum_{k \in A_i} \phi_{ki} a_{kj}.$$

2.1 Arrival Time Uncertainty in Interconnect

Due to a signal transition through an interconnect, arrival times can be quite different between the driver and the receivers of a net. This interconnect delay can be captured as uncertainty of the arrival time for the net. The switching window of a net thus needs to be spread out to cover the uncertainty in interconnect signal propagation. For example, suppose the arrival time is 1000ps at the driver of a net, if the interconnect can take up to the maximum of 200ps to propagate to a receiver of the net, we need to mark the arrival time slots from 1000ps to 1200ns in the switching window of the net. During this time interval, the victims of this net may have crosstalk effects.

Let $d^{\max}_k$ is the maximum interconnect delay on net k. We need to spread out the arrival time as:

$$\begin{cases} a^+_{im} = 1, & \text{if } a_{ij} = 1 \text{ and } j \le m \le j + \lfloor \frac{d^{\max}_k}{s} \rfloor, \\ a^+_{im} = 0, & \text{otherwise,} \end{cases}$$

where a^+_{im} is the modified arrival time slot m for net i. The maximum noise is thus revised as:

$$\max_{0 \le j \le \lfloor \frac{L}{s} \rfloor} \sum_{k \in A_i} \phi_{ki} a^+_{kj}. \tag{5}$$

Eq. 1 needs to be revised as:

$$x^+_i = d^{\max}_i + \max_{k \in D_i}(x_k + d_{ki} + \delta_{ki}).$$

Note that x_k is still unchanged, and the earliest arrival time is not affected.

2.2 Switching Window Density

Compared with the traditional approach using continuous switching windows, the time slot approach can help to reduce the analysis pessimism. The effectiveness strongly depends on the switching windows' density, which can be defined as the ratio of the number of non-zero time slots to the total number of time slots in a switching window. The traditional continuous switching windows have density 1 by definition. This density measure is a very good metric of how effective the time slot approach reduces the pessimism of noise analysis. If the density is close to 0, the switching window tends to be very sparse, and the time slot approach can cut down most of the pessimism in the maximum crosstalk noise.

Slot size	#Violation	S.W. Density
Continuous	352	1.000
400ps	204	0.891
200ps	162	0.842
100ps	131	0.789
90ps	134	0.779
80ps	132	0.755
70ps	130	0.720
50ps	123	0.720
1ps	94	0.430

Table 1: Slot size effect on the number of noise violations.

2.3 Input Timing Uncertainty

Typically, the arrival time of an input pin of a chip is given not as a constant but a bounded timing range. It may represent the timing uncertainty due to the process, voltage, or temperature variation. Let I be the index set of input nets. For $i \in I$, we have

$$\begin{cases} a_{ij} = 1, & \text{if} \lfloor \frac{y_i}{s} \rfloor \leq j \leq \lfloor \frac{x_i}{s} \rfloor, \\ a_{ij} = 0, & \text{otherwise,} \end{cases}$$

where y_i and x_i are the earliest arrival time and the latest arrival time, respectively. The larger the timing uncertainty, the denser the switching switching windows. Therefore, reducing the timing uncertainty at the inputs can increase the effectiveness of our approach.

2.4 Complexity

Given N nets and total M fanouts of nets in a circuit (similar to a direct acyclic graph with N vertices and M edges), the complexity of calculating the arrival time slots is $O(N+M\lfloor \frac{L}{s} \rfloor)$ by Eq. 3. The dominant operation is actually the maximum noise calculation by Eq. 5, which has the complexity $O(NP\lfloor \frac{L}{s} \rfloor)$, where P is the maximum number of aggressors of a net, or equivalently the maximum cardinality of A_i, where $i = 1 \ldots N$.

2.5 Implementation Consideration

Since a_{ij}'s are Boolean variables, we can compact them into a 32-bit integer. a_{ij} can be easily represented by a bit map. The time slot ranges from $\lfloor \frac{y_i}{s} \rfloor$ to $\lfloor \frac{x_i^+}{s} \rfloor$ for a net i. This approach has very efficient memory usage with slight speed penalty.

3. RESOLUTION AND TRUNCATION ERORS

The size of the time slot is an important factor for the analysis accuracy. We test different slot sizes on a small circuit with 8828 nets and 7956 cell instances using $0.25\mu m$ technology, and check the maximum noise peak over 20% of power voltage. Table 1 shows the slot size effect, where the first column shows the slot size, the second column shows the number of noise violations and the third column shows the switching window density. The second row marked with "Continuous" is the traditional continuous switching window method. In general, a smaller slot size can result in fewer noise violations. However, notice that the slot size 90ps generate more violations than the slot size 100ps.

Consider the case in Figure 4, where the time slot sizes are 20ps and 30ps, respectively. Suppose the two aggressors' arrival times are 25ps and 35ps, respectively. If the time slot size 20ps is used, we can align these two aggressors by assumption. If the time slot size 30ps is used, these two aggressors cannot align to create the maximum noise. Therefore, the finer time slot does not always imply the fewer noise violations. However, if the size of a coarser time slot is a multiple of a finer time slot's size, this can be avoided. That is to say, as long as the time slot size is continuously divided evenly, as in Table 2 (1000ps $\rightarrow$ 100ps $\rightarrow$ 10ps), this problem disappears.

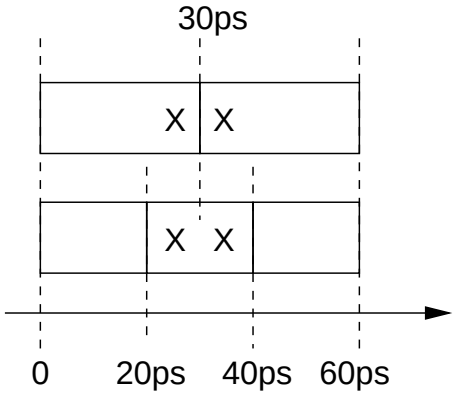

Figure 4: Finer slot gets more noise violations

4. EXPERIMENTAL RESULTS

We conducted experiments on several industrial circuits. Table 2 shows the results. The first column is the circuit name. The second column is the number of nets in that circuit. The third column is the number of cell instances. The fourth column is the time slot size, where "Cont." represents the traditional continuous switching window approach. The fifth column is the switching window density. The sixth column shows the number of noise violations, where we use 40% of VDD as a threshold. The last column shows the run time on a Linux machine with 1.26GHz CPU.

From Table 2, the number of noise violation can be reduced dramatically by 90% for designW and 43% for designA. The run time penalty was just a slight increase. In fact, the continuous switching window approach may take more run time due to the processing of an excess of noise violations. A finer time slot can reduce the number of the noise violations. However, the amount of reduction tends to decrease significantly after some finer time slots.

5. CONSIDERATION OF SLEW RATES

The approach proposed above uses the fastest slew at net i in the computation of ϕ_{ki}. We then assume that this maximum noise effect ϕ_{ki} is achievable in any time slot in the entire switching window. This is a conservative assumption, but a slew at net i may be much larger than the fastest in some time slots, resulting in pessimism. We can incorporate this effect by propagating and maintaining the fastest slew of each time slot and computing ϕ_{ki} of each slot based on the slew there. Let $\alpha_{ij} \in R$ be the minimum or fastest slew in time slot j of net i. The propagated slew model is thus written as:

$$\alpha_{ij} = \min_{k \in D_i} \alpha_{kl}$$

where $l = j - \lfloor \frac{\delta_{ik}+d_k}{s} \rfloor$. The maximum noise is calculated as:

$$\max_{0 \leq j \leq \lfloor \frac{L}{s} \rfloor} \sum_{k \in A_i} \Phi_{ki}(\alpha_{kj})$$

where $\Phi_{k,i} : R \rightarrow R$ is a noise peak function of slew rate, representing the noise effect from the aggressor net k to the victim net i. Note that this approach has heavy speed and space penalty since

Circuit	#Nets	#Instances	Slot size	S.W. Density	#Violations	Run time
designA	19.0K	17.5K	Cont.	1.000	51	16.8s
			1000ps	0.676	41	16.7s
			100ps	0.596	29	17.2s
			10ps	0.470	29	25.4s
designS	25.0K	32.2K	Cont.	1.000	2652	49.8s
			1000ps	0.388	1553	46.1s
			100ps	0.257	1526	46.3s
			10ps	0.058	1507	47.4s
designC	55.8K	50.0K	Cont.	1.000	2815	54.8s
			1000ps	0.785	2210	53.6s
			100ps	0.647	1540	51.7s
			10ps	0.384	1519	54.7s
designB	81.8K	73.0K	Cont.	1.000	7330	176.0s
			1000ps	0.675	4738	176.0s
			100ps	0.596	2951	170.0s
			10ps	0.415	2430	245.0s
designV	273.4K	234.3K	Cont.	1.000	14091	295.0s
			1000ps	0.816	7094	259.0s
			100ps	0.614	3942	237.0s
			10ps	0.596	2831	266.0s
designW	527.1K	444.7K	Cont.	1.000	31782	2118.0s
			1000ps	0.722	10515	1927.0s
			100ps	0.512	3164	1832.0s
			10ps	0.270	3105	1993.0s

Table 2: Comparison of continuous switching window and time slot approach on the number of noise violations.

the bit pattern representation for a_{ij} cannot be used any more and we need to record slew information α_{ij} in each time slot. The number of computations of interconnect and gate delay is proportional to $O(NP\lfloor\frac{L}{S}\rfloor)$. It is not considered practical for multi million-gate design.

6. CONCLUSION

Switching windows can be refined by time slots to improve the accuracy. Our experiments show that up to 90% of potential noise violations detected by continuous switching windows can be excluded by this approach. Moreover, the size of a time slot can be controlled to trade off accuracy to speed and capacity, which makes this algorithm highly scalable.

7. ACKNOWLEDGEMENT

This work is supported by Silicon Perspective, a Cadence company. All of the experiments are done in First Encounter[10] environment and flow. We would like to thank Wei-Jin Dai from Silicon Perspective for his valuable support.

8. REFERENCES

[1] S. S. Sapatnekar. “On the Chicken-and-Egg Problem of Determining the Effect of Crosstalk on Delay in Integrated Circuits”. In *Proc. IEEE of 8th Topical Meeting on Electrical Performance of Eletronic Package*, pages 245–248, 1999.

[2] B. Franzini, C. Forzan, D. Pandini, P. Scandolara, and A. D. Fabbro. “Crosstalk Aware Static Timing Analysis:a Two Step Approach”. In *IEEE of 1st International Symposium on Quality Electronic Design*, pages 499–503, Mar. 2000.

[3] P. F. Tehrani, S. W. Chyou, and U. Ekambaram. “Deep Sub-Micron Static Timing Analysis in Presence of Crosstalk”. In *IEEE of 1st International Symposium on Quality Electronic Design*, pages 505–512, Mar. 2000.

[4] R. Arunachalam, K. Rajagopal, and L. T. Pileggi. “TACO: timing analysis with coupling”. In *Proc. of Design Automation Conference*, pages 266–269, 2000.

[5] P. Chen, D. A. Kirkpatrick, and K. Keutzer. “Switching Window Computation for Static Timing Analysis in Presence of Crosstalk Noise’. In *Proc. of International Conference on Computer Aided Design*, pages 331–337, 2000.

[6] T. Xiao, C. W. Chang, and M. Marek-Sadowska. “Efficient static timing analysis in presence of crosstalk”. In *Proc. of IEEE International ASIC/SOC Conference*, pages 335–339, 2000.

[7] H. Zhou, N. Shenoy, and W. Nicholls. “Timing Analysis with Crosstalk as Fixpoints on Complete Lattice”. In *Proc. of Design Automation Conference*, pages 714–719, 2001.

[8] P. Chen, Y. Kukimoto, C.-C. Teng, and K. Keutzer. “On Convergence of Switching Windows Computation in Presence of Crosstalk Noise”. In *Proc. of International Symposium on Physical Design*, pages 84–89, Apr. 2002.

[9] L. H. Chen and M. Marek-Sadowska. “Aggressor Alignment for Worst-Case Crosstalk Noise”. *IEEE Trans. on Computer-Aided Design*, Vol.20:pp.612–621, May 2001.

[10] “User Guide”. In *First Encounter User Manual, Cadence Design System, Inc.*, Apr. 2002.