

Accurate and Efficient Static Timing Analysis with Crosstalk*

I-De Huang, Sandeep K. Gupta, and Melvin A. Breuer
University of Southern California
{inh, sandeep, mb}@poisson.usc.edu

Abstract

We have developed an accurate and efficient methodology to perform static timing analysis (STA) in combinational logic blocks in the presence of multiple crosstalk-induced noise effects. The crosstalk model used is more accurate because it considers skew, input transition times, and driver strengths. This crosstalk model is enhanced to handle timing ranges for performing STA. The methodology also uses more accurate delay models for gates. The presence of one or more coupling capacitances can create cyclic timing dependencies, even in an otherwise acyclic circuit. We have developed an approach to partition the circuit into minimal timing-iterative subcircuits (TISs) that encapsulate the cyclic timing dependencies. When used in conjunction with our levelization procedure, iterative timing analysis is confined within individual TISs. We have demonstrated that the maximum arrival time values computed by the proposed STA using integrated delay models are much closer to detailed circuit simulation results than an STA that uses the $3C_c$ delay model.

1 Introduction

In recent years, crosstalk noise between adjacent interconnects has become an important concern as feature sizes in CMOS VLSI circuits continue to shrink. Crosstalk gives rise to three types of noise effects, namely pulses, reduction in the arrival and rise/fall times of signal transitions (speed-up), and increase in the arrival and rise/fall times of signal transitions (slow-down). Failure to consider crosstalk effects during timing analysis (TA) may produce delay estimates that are far from the reality. Hence, developing accurate TA tools that consider crosstalk is increasingly important for high performance designs.

Accuracy of static timing analysis (STA) depends upon the delay models used for gates and crosstalk sites. All previous approaches [1][6][7][9] use the less accurate pin-to-pin delay model for gates. Furthermore, as detailed in Section 2, some of these approaches [6][7] use the simplistic $0-C_c-3C_c$ delay model for crosstalk sites. Other

approaches [1][9] use the more accurate AWE [5] approach, but use it in a manner that may cause them to compute optimistic values of worst-case delays. In contrast, the proposed approach uses more accurate models for gates [4] as well as crosstalk sites [12].

The efficiency of an STA approach that considers crosstalk depends a great deal on how it performs TA iteratively to take into account the cyclic dependencies created by the presence of coupling capacitances at crosstalk sites. Previous approaches [3][11] partition a circuit into clusters, where each cluster corresponds to a strongly connected component (SCC) in a graph representation of the circuit. The idea behind identification of clusters is that each cluster captures cyclic timing dependencies that are considered via iterative timing analysis within the cluster. We present an approach to order (or, levelize) the clusters in such a manner that all cyclic timing dependencies are completely contained within individual clusters. Furthermore, we identify clusters that are smaller than SCCs. Both these enhancements help enhance the efficiency of STA by reducing the amount of iteration.

The paper is organized as follows. In Section 2, the implementation and integration of two delay models is discussed. In Section 3, the proposed methodology to perform STA and the algorithms used to partition the circuit into smallest TISs and to perform TA within each TIS are presented. The experimental setup and results are described in Section 4. Finally, the future work and conclusions are presented in Section 5.

2 Integration of Delay Models

Two delay models are used in our approach, one for gates and another for crosstalk sites. The accuracy and the integration of both models are illustrated and the enhancement of Chen's model to handle timing ranges is described.

2.1 Gate Delay Model

The gate delay model proposed in [4] has been demonstrated as being more accurate than the pin-to-pin delay models currently used for STA. This delay model has also been compared with many other recent models and is shown to be more accurate over wide ranges of

* This research is supported by the Semiconductor Research Corporation under contract No. 98-TJ-646.

device sizes, loads, and transition times and skews associated with transitions at gate inputs. It also captures the effects of more variables than table lookup methods can handle. In particular, considering the same input variables as [14], this gate delay model has been shown in [4] to be more accurate than those presented in [15][16].

In addition, given **ranges** of arrival and transition times at each of its inputs, this model facilitates the computation of the minimum and maximum values of arrival and transition times at the output.

2.2 Chen's Crosstalk Model

An analytic model for crosstalk delay analysis was developed and used in [12] to characterize cases where inputs to one or both coupled lines have transitions with arbitrary transition times and directions. The equation describing the voltage of the victim line is shown below.

$$V_{sd}(t) = V_{exp}(t) - \frac{1}{x} \left[\frac{b}{(w+1/x)(w-u)} e^{wt} + \frac{b}{(u+1/x)(u-w)} e^{ut} + \frac{b}{(w+1/x)(u+1/x)} e^{-x} \right] U(t-z),$$

where

$$V_{exp}(t) = 1 - e^{-\frac{t}{y}} - \frac{1}{y} \left[\frac{w+f}{(w+1/y)(w-u)} e^{-wt} + \frac{u+f}{(u+1/y)(u-w)} e^{-ut} + \frac{f-1/y}{(w+1/y)(u+1/y)} e^{-\frac{t}{y}} \right],$$

and $U(t)$ is a unit step function, $b = C_c/(R_a C_t)$, $f = (C_c + C_v)/(R_a C_t)$, $C_t = C_c C_a + C_c C_v + C_a C_v$, and w and u are solutions of impedance characteristic function

$$s^2 + s \left(\frac{R_a(C_c + C_a) + R_v(C_c + C_v)}{R_a R_v C_t} \right) + \frac{1}{R_a R_v C_t} = 0.$$

The parameters x and y are the transition times (i.e., rise/fall times) parameters at the inputs of the drivers of the affecting (A) and victim (V) lines, and z is the skew between these two arrival times at the inputs. If fixed-values are given for parameters x , y , and z , $V_{sd}(t)$ is simply a function of time, and the timing information at V can be easily calculated by using a numerical method. For example, a method such as the Newton-Raphson method can be used to obtain the values of t for which $V_{sd}(t)$ is equal to 0.5 for computation of the delay time, and 0.1 and 0.9 (or 0.2 and 0.8) for the transition time. Computing the minimum or maximum delay from the equation becomes complicated if min-max ranges are given for x , y , and z , as is the case in TA. We will describe in Section 2.4 a numerical approach that enables the use of Chen's model in conjunction with timing ranges.

2.3 Motivation for Using Chen's Model

The $0-C_c-2C_c$ delay model has been used in previous approaches to perform STA when crosstalk effects are to be considered [10]. Let us consider the crosstalk site shown in Figure 1(a). In this approach the coupling capacitance C_c is first considered as extra wire load at A and V , which corresponds to $\alpha = 1$ in Figure 1(b), and the minimum and maximum values of arrival times are computed by performing STA for the whole circuit. The ranges of arrival time computed above are now analyzed to incorporate the effect of crosstalk delay. For example, if

the arrival time ranges for rising transition at A and falling transition at V overlap, then $\alpha = k$, where $k = 2$ for $0-C_c-2C_c$ model and $k = 3$ for $0-C_c-3C_c$ model, is used to recompute the maximum values of the timing ranges at A and V . Also, if arrival time ranges for rising transitions at A and V overlap, then $\alpha = 0$ is used to recompute the minimum values of the ranges at A and V . This is repeated for all possible slow-down and speed-up that can occur at this crosstalk site.

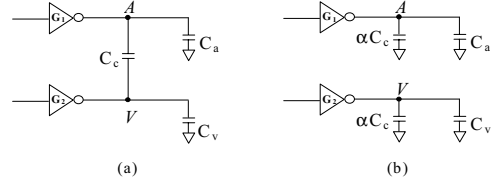


Figure 1. (a) A model of a crosstalk site where C_a and C_v consist of the line capacitance and the gate capacitance in fanouts of A and V , respectively, and C_c is the coupling capacitance between the lines. (b) The model used in $0-C_c-kC_c$ approaches.

The complexity of applying $0-C_c-2C_c$ or $0-C_c-3C_c$ delay models to perform TA is low. At most crosstalk sites and for most values of x , y , and z , these models are very pessimistic. However, even the $0-C_c-3C_c$ -delay model is sometimes optimistic because the delay introduced by crosstalk effect can change by a factor of four or more based on the formulas derived in [2][13].

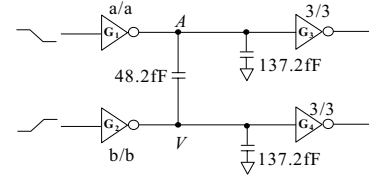


Figure 2. The example circuit for the demonstration of TA accuracy.

We use the circuit in Figure 2 to compare the $0-C_c-2C_c$ delay model, the $0-C_c-3C_c$ delay model, and the crosstalk model used in this paper is shown. The HSPICE results using $0.5\mu\text{m}$ technology are shown in Figure 3. We present the data for three different combinations of driver sizes. For each combination, we assume zero skew and five different values of transition times (in each case, the transition times for A and V are equal). Recall that in the kC_c -delay model, $\alpha = 1$ (see Figure 1) is used to compute maximum delay for both lines if the necessary transitions cannot occur at the same time, i.e., the timing ranges are not overlapped. Otherwise, the min-max values are recomputed using $\alpha = 0$ or k (see Figure 1).

Figure 3 shows the delay trends of these crosstalk models with different driver ratios. Both $0-C_c-2C_c$ -delay and $0-C_c-3C_c$ -delay models compute the same delay for a particular transition time at V , independent of the driver sizes of A , since these parameters are not considered in these models. Hence one can conclude that neither of these

models is suitable for universal use at all crosstalk sites in a circuit, since different sites are driven by gates with different combinations of driver strengths.

With different ratio of strengths of drivers at the crosstalk site, the $0-C_c-3C_c$ delay model usually has the most pessimistic results and the error varies from 4.1% to 56% as shown in Figure 3. However, it is more suitable than the $0-C_c-2C_c$ delay model when the driver size of A is relatively higher. In contrast, the $0-C_c-2C_c$ delay model seems to be less pessimistic (from 28% to -15%) and achieves a better result than the $0-C_c-3C_c$ delay model when the driver sizes of both lines are comparable.

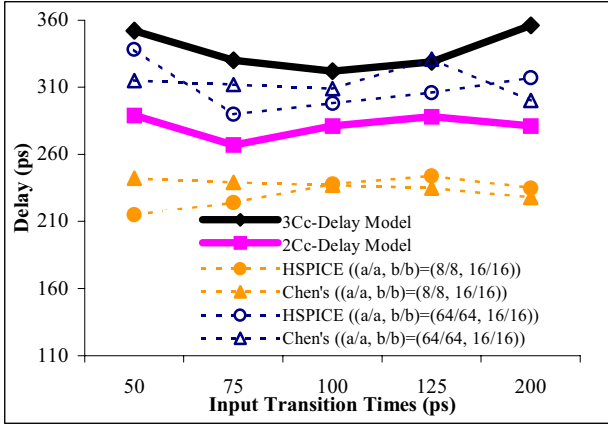


Figure 3. The delays at line V for different crosstalk delay models with different driver ratios and zero skew.

The above experiments might prompt one to conclude that one might use the $0-C_c-2C_c$ model at crosstalk sites with drivers with nearly equal sizes and use the $0-C_c-3C_c$ model at sites with much stronger affecting line drivers. However, the $0-C_c-3C_c$ model can give very pessimistic or optimistic results when the transition times of A and V are different or the skew is not close to zero. Table 1 clearly shows that the $0-C_c-3C_c$ model at sites with much stronger affecting line drivers are too optimistic (-26%) when the skew between A and V is -50ps and the transition times are 200ps.

Skew (ps)	$z = -50$		$z = 0$		$z = 50$	
	100	200	100	200	100	200
Transition Time (ps)	100	200	100	200	100	200
HSPICE (ps)	305	321	298	317	290	296
Chen's Crosstalk Model (ps)	309	299	309	300	308	298
Error percentage (%)	1.3	-6.8	3.7	-5.4	6.2	0.7
$0-C_c-3C_c$ Model (ps)	240*	237*	322	356	240*	237*
Error percentage (%)	-21	-26	8.1	12	-17	-20

Table 1. Comparison of crosstalk slow-down computed by Chen's crosstalk model and $0-C_c-3C_c$ -delay model for different skew values.

* In this model, slow-down occurs only for zero skew. For all other cases, $\alpha = 1$ in Figure 1.(b) is used.

In summary, the above models often over-estimate crosstalk delay. Also, at most typical crosstalk sites, significant slow-down/speed-up can occur even when their appropriate arrival time ranges do not overlap, provided they can have skew values that are within a few gate delays. (In some cases, even the peak slow-down occurs at a small skew.) Completely ignoring slow-down/speed-up if the corresponding arrival time ranges do not overlap can underestimate the worst-case delay.

In contrast, Chen's model gives delay values that are consistently close to the HSPICE results with small error percentages (from -6.8% to 6.2%), over the entire range of skews and input transition times.

2.4 Chen's Crosstalk Model for Timing Ranges

Chen's crosstalk model [12] can be used to obtain the arrival and transition times at the far end of A and V only when fixed values of transition times and arrival times are provided at the inputs of the drivers of A and V . However, in STA, ranges at the far end of crosstalk site must be computed considering crosstalk speed-up and slow-down, given timing ranges at the inputs of the drivers of the site. Next we describe a numerical approach that we have developed that uses gradient methods [8] to find the minimum and maximum values of arrival and transition times to take into account the worst-case delay due to crosstalk slow-down and the best-case delay due to crosstalk speed-up, given the timing ranges at the inputs of the drivers. Our two-part approach is described next.

First, since we are given ranges of values of x , y , and z , we must identify specific combinations of values of these parameters that can give minimum and maximum values of arrival and transition times.

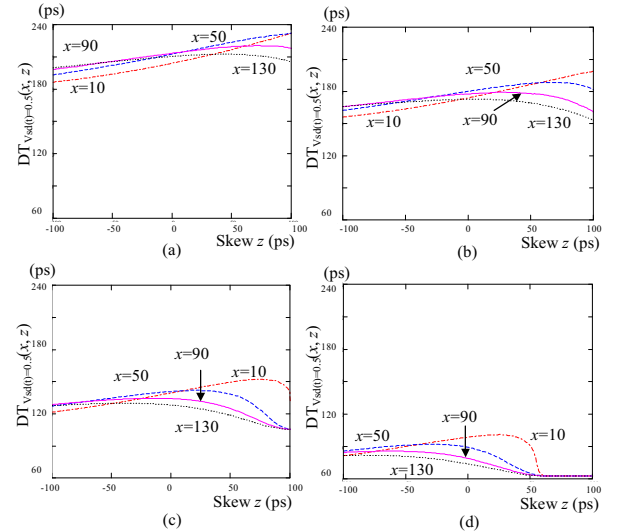


Figure 4. The arrival time of the victim line ($V_{sd}(t) = 0.5$) is bitonic or monotonic for different y values (a) $y = 130$ (b) $y = 90$ (c) $y = 50$ (d) $y = 10$.

Figure 4 illustrates the following observations that we use to simplify this process. (1) The delay at V is bitonic (first increasing and then decreasing) or monotonic with respect to skew z when the timing parameter y is fixed. (This dependence is called $DT_{Vsd(t)=0.5}(x, z)$.) (2) The maximum and minimum delays at V occur when maximum y (y_{max}) and minimum y (y_{min}) are used, respectively. (3) If the range of the skew is infinite, the maximum delay at V occurs when x_{min} and y_{max} are used. However, for a finite range of z , appropriate x and z values must be identified. (4) The minimum delay at V occurs when $y = y_{min}$ and at one of combinations of extreme values of x and z , namely (x_{min}, z_{min}) , (x_{max}, z_{min}) , (x_{max}, z_{min}) , and (x_{max}, z_{max}) .

The above observations are used to perform TA for crosstalk sites. To deal with input timing parameter ranges at the crosstalk site, we use numerical methods to approximate the arrival time (measured when $V_{sd}(t) = 0.5$) and the transition time (measured from $V_{sd}(t) = 0.1$ to $V_{sd}(t) = 0.9$; or 0.2 to 0.8). We use the observation that the function $DT_{Vsd(t)=0.5}(x, z)$ is bitonic or monotonic when one of the parameters, says y , is a fixed value. One of the four combinations, (x_{min}, z_{min}) , (x_{max}, z_{min}) , (x_{max}, z_{min}) , and (x_{max}, z_{max}) , yields the minimum arrival time when y is fixed at y_{min} because of the bitonic or monotonic characteristics. Now, due to the monotonicity with respect to y , the maximum arrival time at V is obtained by fixing y as y_{max} (the maximum value of y) and iteratively changing the values x and z within their ranges. A similar method is used to obtain the min-max values of the transition times of A and V .

Of special importance is the fact that if the skew z has an infinite range, the maximum arrival time at V can be simply obtained by setting x at the minimum value x_{min} and y at the maximum value y_{max} . However, this combination of x and y values does not provide the maximum arrival time for any finite range of skew z . In Figure 4.(c) ($y = 50$), the maximum arrival time of V occurs when x is equal to 50 instead of 10 when the range of the skew (z_{min}, z_{max}) is $(-50, 0)$. In such a case, the approaches in [1][9][11] compute erroneous values of slow-down. Therefore, the above iterative method has to be used to obtain the maximum arrival time of V . The other major observation is that significant slow-down can occur even when the timing ranges of A and V do not overlap. This causes the $0-C_c-kC_c$ method to underestimate crosstalk slow-down at some sites.

Once the appropriate fixed values of x , y , and z are identified for a particular minimum and maximum values calculation, in the second step Chen's formula in Section 2.2 is used to obtain an expression of the voltage at the far end of V (or A) as a function of time. Newton-Raphson's method is then used to compute the value of time for which voltage is 0.5 (for arrival time computation) or values of time at which the voltage becomes 0.1 and 0.9 (for transition time computation).

2.5 Integration of Gate and Crosstalk Models

The above two delay models are used to compute gate delays and to perform STA with multiple crosstalk effects as shown in Figure 5. The new gate delay model is used to calculate the timing ranges at output of a gate (say D_7), given the ranges at its inputs (D_5 and D_6 for D_7). Such a model is used for all gates except the drivers of crosstalk sites. The crosstalk model deals with the timing range calculations from inputs (say D_1 and D_3) of A and V to the far ends of the two lines (D_2 and D_4 for the crosstalk site shown).

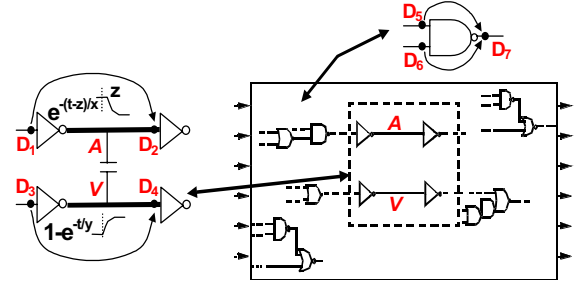


Figure 5. The demonstration of the model integration.

2.6 Acyclic and Cyclic Cases

In a block of combinational logic, the existence of one or more crosstalk sites can give rise to cyclic dependency between the delay values at circuit lines. For example, a single crosstalk site whose one line is in transitive fanout of the other, such as the crosstalk site with lines 20 and 22 in Figure 6, can cause cyclic timing dependency. In general, multiple crosstalk sites can cause cyclic timing dependency, even when none of the crosstalk sites has the above characteristic. For example, in the circuit in Figure 6, the crosstalk with lines 10 and 11 and the crosstalk site with lines 8 and 23 collectively cause cyclic timing dependency. This is the case because the delay value at line 8 cannot be computed without the knowledge of the arrival time range at line 23. The arrival time at line 10 depends on that at line 8, the arrival time at line 11 depends on that at line 10, and the arrival time at line 23 depends upon that at line 11.

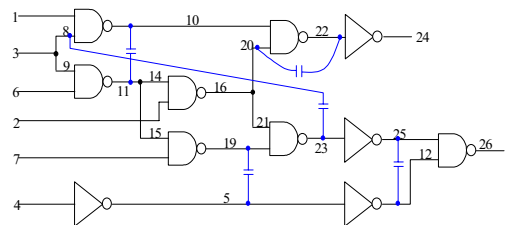


Figure 6. A circuit with multiple crosstalk sites.

3 The Proposed Approach for STA

3.1 Preprocessing and Initialization

We now describe the proposed approach for STA. Our TA begins with the combinational circuit without any coupling capacitance. This circuit is represented as a graph where edges correspond to circuit gates or fanout stems and vertices correspond to circuit lines as shown in Figure 7. This graph can be leveled and classical STA techniques used to compute an initial timing range for each transition (i.e., rising and falling) at each line.

Once the initial timing ranges are computed, a bi-directional edge is added corresponding to the coupling capacitance at each crosstalk site. In general, the resulting graph has cyclic timing dependencies and hence is cyclic.

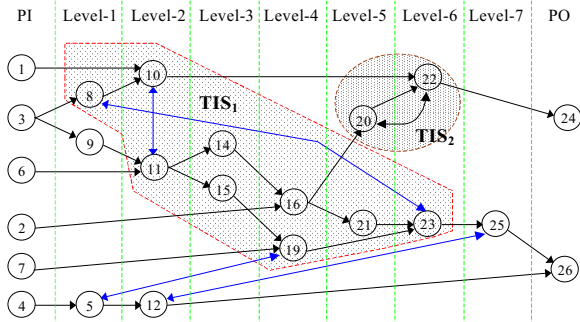


Figure 7. The directed graph for the circuit in Figure 6 with two TISs before re-levelization.

3.2 Timing-Iterative Subcircuit (TIS)

The circuit is partitioned into smallest possible subcircuits, that we call timing-iterative subcircuits (TISs). Each TIS is similar to a strongly connected component (SCC), except that the crosstalk sites that constitute a leaf vertex of SCC are not included in TIS. Hence, the size of TIS is smaller than that of corresponding SCC in [3].

For example, in Figure 7 vertex 5 is not included in TIS₁, since it is not necessary to re-compute the timing ranges of vertex 5 (which represent the timing at input of the driver of a crosstalk site) whenever the timing ranges of vertex 19's inputs (vertex 15 and vertex 7) are updated. The timing ranges at the crosstalk site depend only on the timing ranges of the inputs of the drivers of A and V . The computation of the timing ranges of vertex 5 only need to be performed once if an appropriate levelization is adopted, i.e., the timing computation at vertex 5 is performed after that for TIS₁. Hence, vertex 5 is assigned a level greater than TIS₁.

Each TIS encapsulates cyclic timing dependencies and is subsequently viewed as a single multi-input multi-output gate. In the macroscopic view of the circuit obtained in this manner (see Figure 9 for this view for the circuit in Figure 6 and directed graph in Figure 7), all cyclic timing dependencies are encapsulated within individual TISs. Hence, we only need one pass to perform STA at the macroscopic level if the modified levelization is applied.

3.3 Identification of TISs and Levelization

Our approach, shown in Figure 8, is to partition the circuit into subcircuits such that if each subcircuit is viewed as a single multi-input, multi-output gate, then at the macroscopic level the overall circuit seems acyclic. In other words, all cyclic timing dependencies are encapsulated within individual TISs. To reduce the complexity of iterative timing calculations, it is desirable to find TISs of minimal sizes.

We have developed a modified version of depth-first-search to partition the given circuit into TISs of minimal sizes. This algorithm partitions the circuit into TISs of minimal sizes that are smaller than SCCs.

Before performing the modified depth-first-search (DFS), all vertices are marked unvisited and all primary outputs are marked as the stop points (SP). During DFS, a vertex is marked as a stop point (SP) if all possible paths beyond this vertex have been searched. (This reduces the complexity of DFS.) DFS is executed starting at each primary input; each DFS backtracks when it reaches one of the primary outputs or a SP.

During the search, a cyclic path is found if the current vertex is found as having been visited earlier. All the vertices from the current vertex back to the repeated vertex along this path are examined. If any of these vertices is already assigned to a TIS, then all these vertices are assigned to the same TIS. Otherwise, they are all assigned to a new TIS. In this manner, the procedure identifies minimal TISs. The complexity of the search procedure is $O(V+E)$, where V and E are the number of vertices and edges respectively.

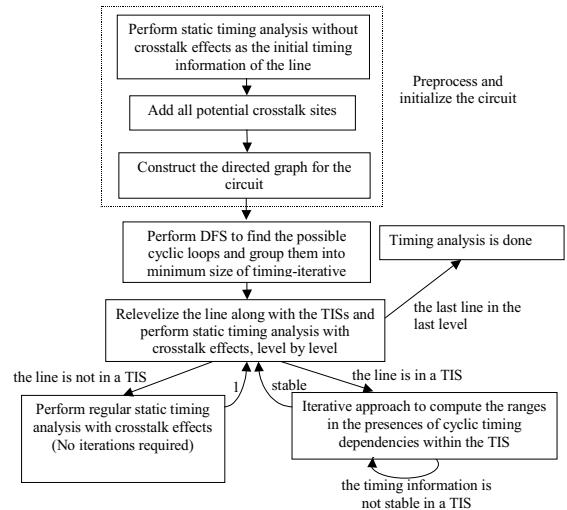


Figure 8. Proposed approach for STA with crosstalk effects.

For example, in the graph shown in Figure 7 all primary outputs, namely vertices 24 and 26, are marked as SP. One of the primary inputs, say vertex 1, is first selected to start the search. Therefore, the path $1 \rightarrow 10 \rightarrow$

22 → 20 → 22 is found that revisits vertex 22. A new TIS number is assigned to vertices 20 and 22, since vertex 20 is examined and it does not belong to any TIS. Vertex 22 is marked as SP because no more paths need to be searched. The search continues in this manner until all vertices are marked as SP.

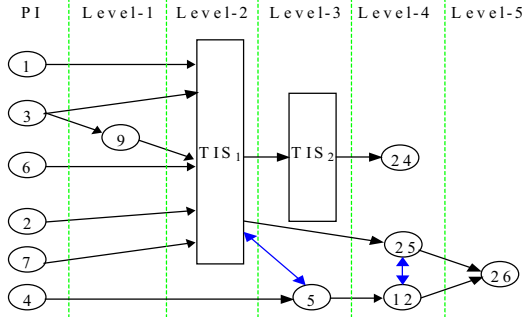


Figure 9. The macroscopic view of the circuit in Figure 7, after each TIS is replaced by a multi-input, multi-output block.

Once the TISs are identified, each TIS is viewed as a single multi-input, multi-output gate to obtain a macroscopic view of the circuit. Subsequently, our levelization is applied to relevelize the circuit to obtain the macroscopic view and levels. One key difference in our levelization of the macroscopic view of a circuit occurs at crosstalk sites that are not contained within any TIS. In such cases, once the levels at the drivers of A and V lines are known, the maximum of these two values plus 1 is assigned as the level of A as well as V . This ensures that the timing ranges of both input drivers of a crosstalk site are computed before the timing ranges at the crosstalk site are computed. Therefore, one iteration for performing STA at the macroscopic level is guaranteed if the above technique is applied. The directed graph of the circuit and TISs shown in Figure 7 obtained in this manner is shown in Figure 9.

At the macroscopic level illustrated in Figure 9, only one iteration is needed for TA. Multiple iterations may be required within each TIS, however. For each TIS, the original levels of the vertices are used to perform event-driven TA level-by-level.

3.4 Iterative Timing Analysis

When STA with crosstalk effects is performed, the intra-TIS timing simulation is triggered if the timing range at one of the lines in the TIS is updated. TA in the TIS will follow the levels assigned to the lines in the original circuit. When the second iteration of TA begins, the change in arrival time at lines can only change the skew calculation (i.e., parameter z in the equation) for a crosstalk site, and the previous timing range is updated only if the result being changed in skew yields a new maximum or minimum arrival time. The analysis stops

when min-max values of all lines in the current level do not change, i.e., when timing ranges of all lines in the TIS are stable.

3.5 Convergence

The proof of convergence of our approach depends primarily upon bitonic or monotonic characteristics illustrated in Figure 4. The iterative timing analysis approach to compute the timing ranges of lines within TISs only changes/updates the skew for the crosstalk site, and keeps the transition time unchanged after the first run of TA within the TIS. Hence, the only variable after the first pass is skew z . Note that, in any useful circuit, each line has a finite range of arrival time. Since the arrival times are only updated when the computed value is larger (smaller) than the current maximum (minimum), eventually the finite maximum arrival time within this range of the skew will be reached. Once the arrival times become stable, the range of z becomes stable, and TA is terminated.

4 Experimental Results

In this section, TA is performed for ISCAS85 benchmark circuits. All gates in the benchmark circuits are assumed to be minimum-size transistors, i.e., with width 3λ , except the drivers of the crosstalk sites, where the N and P transistors have widths 16λ . The resistance and capacitance of the gates are computed based on the parameters in 0.5 μm technology, and the coupling capacitances are extracted from layouts.

In the first experiment, STA without crosstalk is performed to obtain the timing ranges as shown in the first three columns of Table 3. The proposed methodology is then used to find TISs of minimal sizes for the circuit. Table 2 shows the number of TISs found and the corresponding numbers of fanins, fanouts, the number of coupled lines included in TIS_i (LC_i), the number of extra lines included in TIS_i (EL_i), and internal nodes in TIS_i (L_i). The proposed approach for TA needs less than five iterations to compute the timing ranges for most TISs. However, it takes 12 iterations to compute the timing ranges for one of the TISs in C1355, because 568 lines are encapsulated within this TIS. The iteration overhead (IOH) for computing timing ranges considering crosstalks is defined as

$$IOH = \frac{K * W_1 + LCX * W_2 + \sum_i (LC_i * W_2 * N_i + LK_i * W_1 * N_i)}{K * W_1 + \sum_i (L_i * W_1)} - 1,$$

where K is the number of the lines not included in any TIS, LCX is the number of the coupling lines outside TISs, and N_i is the number of iterations performed in TIS_i .

If the average computation complexities at crosstalk and non-crosstalk lines are five and one units, respectively, $IOH_{C432} = ((38*5+151)*5+(2*5+1)*2+216+34*5)/432-1 = 3.86$ for the circuit C432, and only 0.16 for C2670. IOH

depends mainly on the number of lines in the TIS and the number of iterations. Therefore, IOH can be reduced by reducing the number of lines in the TIS, the number of iterations for each TIS, and even the number of TISs.

ISCAS 85 Circuits	Number of Crosstalk Sites	Number of TISs	Number of Fanins for Each TIS	Number of Fanouts for Each TIS	L_i	LC_i	N_i	EL
C17	2	0	0	0	0	0	1	0
C432	25	2	53	30	189	38	5	0
			7	5	3	2	2	
C880	36	3	40	45	83	19	3	2
			4	3	5	3	2	
			1	9	3	2	2	
C1355	88	3	99	70	568	110	12	1
			7	6	6	3	2	
			3	2	3	2	2	
C1908	34	0	0	0	0	0	1	0
C2670	48	1	11	21	20	5	2	1
C3540	71	0	0	0	0	0	1	0
C5315	105	0	0	0	0	0	1	0
C7552	151	2	36	127	98	36	5	4
			2	10	5	2	3	

Table 2. Details of the TISs obtained for benchmark circuits.

Table 3 illustrates that the min-max timing ranges of the circuit with crosstalk effects are wider than those without crosstalk effects. For example, C880 contains four significant coupling capacitances along a critical path with capacitances between 12.8fF and 5.7fF. (A total of 36 crosstalk sites are considered for this circuit.) This increases the range on the arrival time (i.e., the difference between maximum and minimum arrival times) from 9.373 to 10.327ns, and the maximum arrival time by $(12.093-11.139)/11.139 = 8.56\%$. As can be seen, the increased percentage of the maximum arrival time is very small in C1908. This is because only one coupling capacitance is located on a critical path. For some circuits, the minimum arrival time with crosstalk effects is slightly larger than that without crosstalk effects. This is because the extra load is added at the crosstalk site when the coupling capacitance is added.

Timing ranges using the $3C_c$ -delay model are also shown in Table 3. For the circuit C880, $3C_c$ -delay model compute maximum arrival time at the primary output that is 8.4% higher than the proposed approach with integrated model. In cases where several crosstalk sites are located along critical paths and have larger coupling capacitance values, the $0-C_c-3C_c$ model overestimates the maximum delay by a large percentage. For example, it overestimates this value by 44.6% for C17, which has coupling capacitance 73fF. In contrast, the overestimate is only 0.3% for C1908, since in this circuit only one crosstalk site with a coupling capacitance 17.7fF along a critical path.

The difference between the maximum arrival time computed for the $3C_c$ -delay model and the crosstalk model that we use can be negative, because only $1C_c$ -delay is considered for $3C_c$ -delay model if the arrival timing

windows of A and V do not overlap. In that case, $3C_c$ -delay model will under-estimate the maximum arrival time. However, this was not observed in the experiments reported here.

ISCAS85 Circuits	Static Timing Analysis Without Crosstalk		Integrated Model With Crosstalk		3C _c -Delay Model With Crosstalk		The Over-estimate of the Maximum arrival time (%)
	Minimum arrival time (ns)	Maximum arrival time (ns)	Minimum arrival time (ns)	Maximum arrival time (ns)	Minimum arrival time (ns)	Maximum arrival time (ns)	
C17	0.228	0.760	0.285	1.844	0.454	2.667	44.6
C432	0.530	11.305	0.530	12.875	0.530	13.929	8.2
C880	1.766	11.139	1.766	12.093	1.766	13.111	8.4
C1355	0.707	13.144	0.720	15.427	0.707	16.848	9.2
C1908	0.946	15.907	0.946	16.192	0.946	16.248	0.3
C2670	1.986	18.294	1.986	18.630	2.084	19.149	2.8
C3540	2.425	21.917	2.425	21.943	2.425	22.349	1.9
C5315	1.622	19.308	1.621	20.051	1.643	20.191	0.7
C7552	2.816	16.022	2.816	17.556	2.837	19.202	9.4

Table 3. The results of STA.

The last experiment was performed to illustrate the tightness of the delay computed by our approach. The 36 crosstalk sites used in STA reported in Table 2 (whose coupling capacitances vary from 16.1fF to 1.5fF) are inserted into C880. A critical path is selected and a partially specified pair of patterns that satisfy the necessary conditions to sensitize this path is generated. Then we generate and assign random values to PIs with completely specified values, to obtain either 00, 01, 10, or 11 at each input. This process is then repeated for several critical paths. 1,024 such pairs of patterns are generated in this manner, each of which causes a transition along a critical path. HSPICE simulation as well as timing simulation using our integrated model are performed for each pair of patterns.

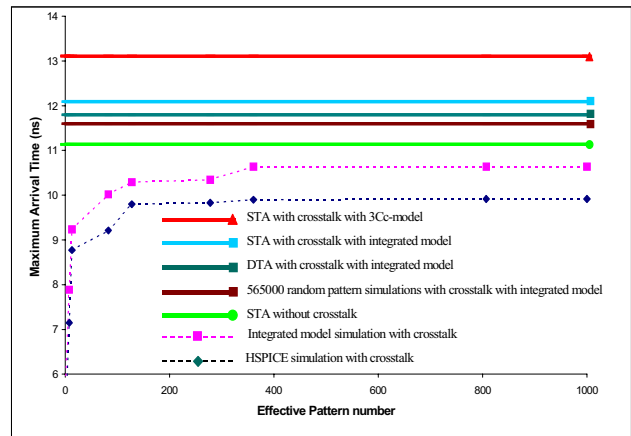


Figure 10. The maximum of arrival time for C880 with and without crosstalk effects.

Figure 10 shows the maximum arrival time obtained by HSPICE simulation for the above 1024 vectors for C880. We also use a version of our program that performs timing

simulation using our integrated delay models. The maximum arrival times computed by the two simulations are fairly close. This shows that the integrated delay model is reasonably accurate. On the same graph, we also show the maximum arrival time computed by using STA that uses our gate model but does not consider crosstalk, uses our integrated model and considers crosstalk, and uses our gate model in conjunction with the $3C_c$ model and considers crosstalk.

Figure 10 also shows the results of our simple heuristic dynamic timing analysis (DTA) and random pattern simulations (RPS) using our integrated model. 565,000 fully specified pairs of patterns are randomly generated and simulated for both cases. The heuristic DTA obtains a higher delay (11.842ns) than RPS (11.588ns) for the same number of random patterns.

In summary, Figure 10 shows that the STA methodology described in this paper is significantly more accurate than the previous STA methodologies that consider crosstalk. Also, they provide motivation for our ongoing work on DTA.

5 Conclusion

We have developed an accurate and efficient methodology to perform STA in combinational circuits in the presence of multiple crosstalk-induced noise effects. The crosstalk model used in this paper is more accurate because it considers skew, input transition times, and driver strengths. This crosstalk model is enhanced to handle timing ranges for performing STA. The methodology also uses more accurate delay models for gates. Our methodology efficiently performs STA when multiple crosstalk effects exist that create cyclic timing dependencies. Our STA approach partitions the circuit into TISs of minimum size. Each TIS encapsulates cyclic timing dependencies and all cyclic timing dependencies are completely encapsulated within individual TISs. We use a leveled approach to process gates as well as TISs during STA, where TISs, in general, require multiple iterations.

To further reduce complexity, a bi-directional edge introduced by a coupling capacitance can be replaced by a uni-directional edge, if the driver of one line of a crosstalk site is much stronger than that of the other line in the site. The direction will be from the line with the stronger driver to the line with the weaker driver. This is due to the fact that for crosstalk sites with such asymmetric drivers, the timing changes very little at the line driven by the stronger driver. By doing this, the size of TISs and the complexity of TA are both reduced. In some cases, this also reduces the number of TISs.

We have demonstrated that the maximum arrival time values computed by the proposed STA using integrated delay models are much closer to detailed circuit simulations than STA that uses $3C_c$ delay model. We also demonstrate the need for performing DTA to obtain tight

upper bounds on delay values, especially when crosstalk is considered.

References

- [1] F. Dartu and L. T. Pileggi, "Calculating Worst-case Gate Delays Due to Dominant Capacitance Coupling", *Proc. Design Automation Conf.*, pp. 46-51, 1997.
- [2] H. Kawaguchi and T. Sakurai, "Delay and Noise Formulas for Capacitively Coupled Distributed RC Lines", *Proc. Asia and South Pacific Design Automation Conf.*, pp. 35-43, 1998.
- [3] H. Zhou, N. Shenoy, and W. Nicholls, "Timing Analysis with Crosstalk as Fixpoints on Complete Lattice", *Proc. Design Automation Conf.*, pp. 714-719, 2001.
- [4] L. C. Chen, S. K. Gupta, and M. A. Breuer, "A New Gate Delay Model for Simultaneous Switching and Its Applications", *Proc. Design Automation Conf.*, pp. 289-294, 2001.
- [5] L. T. Pileggi and R. A. Rohrer, "Asymptotic Waveform Evaluation Timing Analysis", *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, pp. 352-366, 1990.
- [6] P. Chen, D. A. Kirkpatrick, and K. Keutzer, "Miller Factor for Gate-Level Coupling Delay Calculation", *Proc. ICCAD.*, pp. 68-74, 2000.
- [7] P. Chen, D. A. Kirkpatrick, and K. Keutzer, "Switching Window Computation for Static Timing Analysis in Presence of Crosstalk Noise", *Proc. ICCAD.*, pp. 331-337, 2000.
- [8] P. R. Turner, "Guide to Numerical Analysis", *PR Macmillan Education Ltd.*, pp. 140-146, 1990.
- [9] R. Arunachalam, K. Rajagopal, and L. T. Pileggi, "TACO: Timing Analysis with Coupling", *Proc. Design Automation Conf.*, pp. 266-269, 2000.
- [10] S. S. Sapatnekar, "A Timing Model Incorporating the Effect of Crosstalk on Delay and Its Application to Optimal Channel Routing", *Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 19, No.5 pp. 550-559, 2000.
- [11] T. Xiao and M. Marek-Sadowska, "Worst Delay Estimation in Crosstalk Aware Static Timing Analysis", *Proc. International Conf. on Computer Design*, pp. 115-120, 2000.
- [12] W. Y. Chen, S. Gupta, and M. Breuer, "Analytic Model for Crosstalk Delay and Pulse Analysis Under Non-Ideal Inputs", *Proc. International Test Conf.*, pp. 809-818, 1997.
- [13] Y. Sasaki and G. D. Micheli, "Crosstalk Delay Analysis Using Relative Window Method", *Proc. ASIC/SOC Conf.*, pp. 9-13, 1999.
- [14] V. Chandramouli and K. A. Sakallah, "Modeling the Effects of Temporal Proximity of Input Transitions on Gate Propagation Delay and Transition Time", *Proc. Design Automation Conf.*, pp. 617-622, 1996.
- [15] Y. H. Jun, K. Jun, and S. B. Park, "An Accurate and Efficient Delay Time Modeling for MOS Logic Circuits Using Polynomial Approximation", *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, pp. 557-575, 1999.
- [16] A. Nabavi-Lishi and N. C. Rumin, "Inverter Models of CMOS Gates for Supply Current and Delay Evaluation", *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, pp. 1271-1279, 1994.