Functional Correlation Analysis in Crosstalk Induced Critical Paths Identification

Tong Xiao
Sun Microsystems, Inc.
San Antonio Road
Palo Alto, CA 94303-4900
xiaotong@ieee.org

Malgorzata Marek-Sadowska
Dept. of Electrical & Computer Eng.
University of California, Santa Barbara
Santa Barbara, CA 93106
mms@ece.ucsb.edu

ABSTRACT

In deep submicron digital circuits capacitive couplings make delay of a switching signal highly dependent on its neighbors' switching times and switching directions. A long path may have a large number of coupling neighbors with difficult to determine interdependencies. Ignoring the mutual relationship among the signals may result in a very pessimistic estimation of circuit delay. In this paper, we apply efficient functional correlation analysis techniques to identify critical paths caused by crosstalk delay effects. We also discuss applications to static timing optimization. Experiments demonstrate efficacy of the proposed technique.

1. INTRODUCTION

In digital circuits manufactured in deep submicron technologies, interconnect coupling and intrinsic capacitance are comparable. This may cause delays of switching signals to be highly dependent on switching times and switching directions of their coupled neighbors, and it may also create noise signals on coupled neighbors [11]. These effects are usually referred to as *crosstalk*. When all coupled neighbors of a given wire are quiet, we call it the *nominal* case. If all the neighbors switch in the same direction, *speedup* effect occurs and delay of the given wire may be much smaller than in the nominal case. If all the neighbors switch in the opposite direction, *slowdown* effect occurs and delay of the given wire may be much bigger than in the nominal case. To manage crosstalk effect, efficient static timing analysis is needed.

The accuracy of delay estimation can be improved by incorporating functional information. Functional analysis proposed in [9] identifies pairs of signals in a circuit which are not mutually sensitive. Satisfiability (SAT) formulation is proposed in [4] to find vectors causing maximum peak noise on a given wire. These techniques are not applicable directly in static timing analysis because of complexity issues. The number of paths may be exponential in the number of gates, and there are millions of gates in the modern digital circuits. Besides, a path may have many coupling neighbors creating an extremely large search space for a SAT problem.

In our earlier research we have found that the longest path delays in circuits with capacitive couplings are on average about 20% larger than nominal case delays [13]. Crosstalk induced delay highly depends on switching times of coupling neighbors, and our preliminary results show that the bound on crosstalk caused delay is tighter if we apply efficient techniques to analyze functional correlations between coupling neighbors [14]. Thus gate sizing utilizing more accurate analysis tools becomes an attractive technique to apply in post routing optimization to reduce crosstalk delay effect. In this paper we apply efficient functional correlation analysis techniques to identify critical paths caused by crosstalk delay effects, and also discuss how it can be combined in static timing optimization, using gate sizing as an example.

The rest of the paper is organized as follows. In Section 2 we introduce the necessary background information on static timing analysis in the presence of crosstalk. In Section 3 we discuss critical path identification using correlation analysis. In Section 4 we discuss functional analysis in the context of gate sizing as a technique to reduce crosstalk induced delay. Experimental results are presented in Section 5.

2. BACKGROUND

Depending on the accuracy requirements, different delay calculation methods or even numerical simulations can be applied to compute delays in the presence of crosstalk. We have proposed a simple two- pole model in [12]. It can be applied to compute the necessary parameters before determining the earliest and latest arrival times in a network with couplings.

Multiple Aggressor Worst Delay (MAWD) problem of computing the earliest and latest arrival times of a signal in crosstalk aware static timing analysis is formulated as follows: given victim's and aggressors' input timing windows and input slew rate ranges, compute the victim output's timing window.

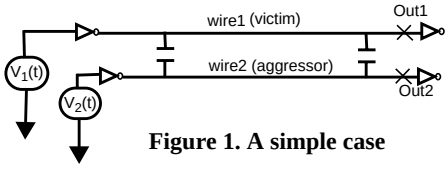


Figure 1. A simple case

This problem has been discussed in detail in our previous work [13]. Here we only give a brief overview. In Fig. 1 we show a simple case of an interconnect and one neighbor. When computing delay at the node Out1 of wire1, wire2 is an *aggressor*, and wire1 is a *victim*. The *victim* and *aggressor* attributes are interchangeable.

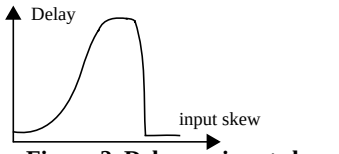


Figure 2. Delay vs. input skew

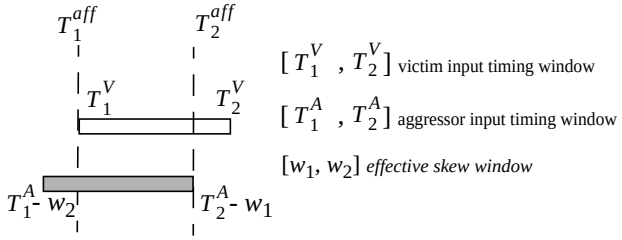
If wire2 and wire1 switch in the opposite directions, delay at Out1 may be much bigger than in the nominal case, i.e., when wire2 is quiet. The difference in delay in these two cases is called a *slowdown*. The difference of switching time between victim's and aggressor's input is the *input skew.* Fig. 2 illustrates how delay at Out1 changes with respect to input skew when neighbors switch in opposite directions. The range of input skews causing at least 5% slowdown at Out1 is called the *effective skew window.*

In [13], it has been shown that the effective skew window and the worst case slowdown caused by an aggressor can be found by checking two corner cases: victim is the fastest and aggressor is the slowest; and victim is the slowest and aggressor is the fastest.

We have also made a pessimistic assumption, that the maximum slowdown by each aggressor occurs when aggressor's input skew is anywhere in the effective skew window. This assumption greatly

*This work was supported in part by the National Science Foundation grant CCR-9811528 and in part by the California MICRO Program through Conexant and Synopsys.

simplifies the MAWD problem. To solve the MAWD problem, we first find the *affected window* for each aggressor, i.e. the switching time of victim input during which the maximum slowdown caused by the aggressor occurs. It can be computed from victim's and aggressor's input timing windows and the effective skew window of this aggressor, as shown in Fig. 3.



Affected window: $T_1^{aff} = \max(T_1^V, T_1^A - w_2)$

$T_2^{aff} = \min(T_2^V, T_2^A - w_1)$

Figure 3. Computing affected window

Next we sort the boundaries of all the affected windows, and sweep the boundaries from left to right. If the sweeping line intersects the affected window i, slowdown effect from the aggressor i is considered. As shown in Fig. 4, when sweeping line is at position 1, we consider slowdown effects from all three aggressors. When sweeping line is at position 2, we only consider slowdown caused by aggressor 2. We only check the boundary points when computing victim's latest arrival time. Sorting takes nlog(n), where n is the number of aggressors of a victim, and sweeping takes linear time. The same strategy is applicable to finding victim's earliest arrival time considering speedup effects.

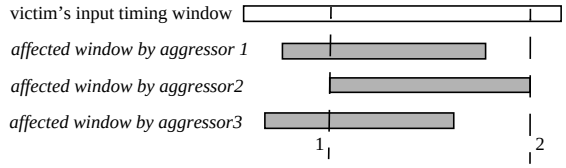


Figure 4. Sweeping affected windows

The above strategy is effective and efficient in static timing analysis. *Simple worst case* approach which assumes that coupling effects may occur at any time, gives a very pessimistic estimation of the longest path delay, on average 70% more than the nominal case delay. Solving the MAWD problem described above results in a bound which in average is 22% over the nominal case delay.

We have shown in [14] that iterative topological analysis is necessary if signals on the same path are mutually capacitively coupled. And our experiments show that convergence usually occurs in two iterations after simple worst delay of each signal is computed.

3. CRITICAL PATHS IDENDIFICATION USING FUNCTIONAL CORRELATION ANALYSIS

Consider a physical path {b, d, e, g} shown in Fig. 5. *On inputs* are signals on the path. *Side inputs* are the fanin signals to the gates along the path but themselves not on the path. For example, c is a side input of signal d. *Neighbors* of an on input are signals routed next to it in layout of the circuit. A logic value is a *controlling value* (*cv*) for a gate if and only if it determines the gate's output independently of other inputs. For example, the controlling value of an AND gate is 0, and the controlling value of an OR gate is 1. The *noncontrolling value* (*ncv*) for a gate is the complementary value of its *cv.*

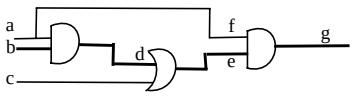


Figure 5. An example of a path

Critical paths are those paths whose delay is longer than given specification. We call gates on critical paths the *critical gates*. There are different methods to identify critical paths when we need to consider crosstalk delay effect, explained as follows from the simplest to most complicated:

1. Selecting paths based on negative slacks:

Slack of each signal can be computed from the required and arrival times. When these times are computed accurately, a path which consists only of nodes with negative slacks are identified as critical. This is the fastest way to determine critical paths. Because of computational complexity, arrival and required times are usually computed based on topological analysis, and provide only a lower bound of the actual slack. Thus this method may report pessimistically large number of critical paths.

2. Selecting paths based on coupling delay:

For each path in the circuit, compute coupling delay at each stage as explained in Section 2, report path whose coupling delay is longer than timing specification. Because the inaccuracy introduced by using estimated required time is removed, this method will report less pessimistic results as compared to the first method. But still no functional information of gates is considered.

3. Selecting paths based on coupling delay and sensitizability:

In this method, only functional irredundant paths whose coupling delay is bigger than specification are selected. The path selection method proposed in [3] can be used.

4. Selecting paths based on coupling delay, sensitizability and functional correlations between coupling neighbors:

In this method, for each functional irredundant path, we analyze the functional correlations between coupling neighbors, and recompute the coupling delay of this path. Utilizing functional correlation information can reduce the pessimism in estimating path delay, and the number of critical paths selected will be much less than in the previous methods. We will explain the details later in this section.

5. Identifying long paths by simulation:

Exhaustive simulation of a chosen path under various input vectors gives the most accurate path delay and may supply the information about criticality under specific input vectors. Such simulation of each path in the circuit is too expensive and impractical, thus usually only a small subset of paths selected by one of the above methods will be simulated for a set of input vectors.

Our target is to identify critical paths for static timing optimization, thus we will not discuss the simulation and test vector generation issues in this paper. Before we explain how to identify critical paths using functional correlation analysis, we first introduce the following definitions.

False paths are those that do not propagate transitions to primary outputs, and thus do not determine circuit's performance. Identification of all false paths is computationally expensive. A necessary condition for a path to be a true path is as follows [5]:

Definition 1. Functional irredundant path: if an on input settles at a noncontrolling value, its corresponding side inputs must also settle at noncontrolling values.

A physical path is treated as two logical paths: one for rising and one for falling transition at the primary input. Path sensitization criterion can be checked by extending logical paths gate by gate until primary output is reached. Because of the complexity of sensitization problems, we only perform implication to check sensitizability. If no conflicts are found, we assume that the path is sensitizable.

Although some signals always have the same type of correlations no matter which input vectors are applied, or which paths are sensitized, path dependent correlations occur more frequently. We have the following definitions to describe such functional correlations.

Definition 2. If two neighboring signals A and B always settle to the same value when the path P containing signal A propagates a transition, we call signal B a *path P dependent obedient neighbor* of A.

Definition 3. If two neighboring signals A and B always settle to opposite values when the path P containing signal A propagates a transition, we call signal B a *path P dependent disobedient neighbor* of A.

Definition 4. If a coupling neighbor is neither path dependent obedient, nor disobedient, we call it an *active neighbor.*

Identifying path dependent obedient or disobedient neighbors is performed after sensitization check. When implications are used to verify if a given path P is sensitizable or not, some signals may also have implied values. For a signal a_i on a sensitizable path P, we check the implied values of a_i's coupling neighbors. If a neighbor has the same value as the signal a_i, this neighbor is path dependent obedient neighbor; if it has an opposite value to signal a_i, it is a path dependent disobedient neighbor. In this way we can find some path P dependent obedient and disobedient neighbors.

After identifying path dependent correlations, we apply the following Lemma to compute more accurate path delay.

Lemma 1. For a given path P propagating a transition, its path dependent obedient neighbors will not cause slowdown; similarly, path dependent disobedient neighbors will not cause speedup on P.

With a huge number of paths in modern digital circuits, traversing each path is too time consuming. We use esperance and incremental sensitization checking to prune search in the early stages. *Esperance* of a partial path is the delay of the longest path which contains the partial path. LSP denotes the stored long sensitizable paths. PPS denotes the stored, still investigated, partial paths. A partial path is extended one gate at a time, esperance is computed and sensitization checking is incrementally applied after each extension. The short paths and unsensitizble partial paths are removed from PPS.

The following procedure is used to find all critical paths when timing specification τ is given:

Procedure ***find_critical_paths***

LSP = empty; PPS = empty;
For each primary input:
Initialize partial path with value = 1 into PPS;
Initialize partial path with value = 0 into PPS;
While (partial path P<- top of PPS)
If P is not sensitizable:
Remove from PPS, continue;
If P is a complete path (reaches PO):
Compute path coupling delay Cd;
If Cd >= τ, P->LSP;
Remove P from PPS;
Else:
Expand P into a new partial path P1
by picking a remaining fanout;
If esperance of P1>= τ, insert P1->PPS;

Note that we can apply the above procedure to all the path in the circuit, but we can also choose to apply it only to some paths we are interested in. This feature makes it attractive to use in timing optimization, as we will explain in Section 4.

Functional correlation analysis in comparison to the purely topological analysis, gives less pessimistic estimation of path delay, and identifies much fewer paths as critical. It is different from false path analysis. False paths analysis can be done before static timing analysis, and then false paths can be removed from timing graph when performing the analysis. Correlation analysis may need to be performed more than once, as it may be too expensive to store all the correlation information when memory is limited.

4. DISCUSSION OF APPLYING FUNCTIONAL ANALYSIS IN GATE RESIZING

We propose to perform gate resizing after layout is done. This is a necessary step, as we need to adjust gate sizes to reduce coupling delays. Down sizing can be performed to reduce power when timing specification is met. We propose to first perform gate sizing to make sure that nominal case delay of the circuit is close or equal to the given timing specification, and then readjust gate sizes to reduce coupling delay.

The reason for this strategy is that delay calculation and timing analysis considering crosstalk is more memory- and CPU-intensive than the traditional timing analysis, which in turn makes timing optimization considering crosstalk more time consuming.

The general gate sizing strategy proposed in [6][7][8] can be adopted and extended to reduce coupling delay. Here we only discuss the following two issues:

1. Compute slack for each signal based on topological analysis:

In static optimization, slack of each signal needs to be computed. The earliest and latest arrival time can be computed as explained in Section 2.

Computing accurate required time for each signal is complicated because delays depend on neighbors switching times. The problem is formulated as follows:

Coupling Aware Required Time Computation (CARTC): Given the required time $T_{required}$ on a signal, and arrival times on all its coupling neighbors' inputs, compute the required time on the given signal's input $T_{in\_required}$.

The following iterative method can be applied to solve the CARTC problem:

(1) Assume the delay t_{stage} from input of this signal as simple worst case delay, i.e., worst slowdown effect from each aggressor occurs. $T_{in\_required} = T_{required} - t_{stage.}$

(2) Compute delay t_{stage} when input arrival time is equal to $T_{in\_required}$ using the method of solving MAWD problem as explained in Section 2. Update $T_{in\_required}$ using the new t_{stage}.

(3) Iterate step 2 until convergence.

Note that there is a trade-off here: if we apply only step 1 to compute required time, it always gives a pessimistic estimation, and more signals in the circuit may have negative slacks. The benefit is that we only need to recompute required time if there is a change in the fanin or fanout cone. But if we compute more accurate required times by iteration, they depend on neighbors' switching times. So if neighbors' switching times change during the process of gate sizing, we need to recompute the required time of the given signal.

2. Incorporating Functional Correlation Analysis

Functional correlation analysis can be applied to compute more accurate arrival times of signals, thus more accurate slacks.

It can also be applied after gate sizing to perform more accurate timing verification and help the gate sizing procedure to terminate earlier. The number of times to apply functional correlation analysis can be adjusted according to the number of critical paths in the circuit.

If we can afford to store all correlations for each critical path, it will save us the time of redoing functional analysis. This is only

possible when the number of critical paths is manageable with available computer memory. When the number of critical paths is large, it is also too expensive to perform path based calculation when computing gradient and slack changes in gate sizing.

Gate sizing may not necessary meet the given timing specification. When this occurs, buffer insertion [1][2] or spacing, or even rip-up and rerouting, can be applied to reduce coupling delay.

5. EXPERIMENTAL RESULTS

In our experiments we have extracted resistance and capacitance of some coupling structures in 0.35µm technology and added delay information to the synthesized and mapped logic netlists based on the data derived from these laid out structures. In our experiments, logic stages may have delay from 50ps to 800ps. Slew rates can be 40ps to 400ps. Each wire has up to 3 randomly selected coupling neighbors. Depending on the delay of the coupling neighbor, contribution to coupling delay of each neighbor may be up to 45% of the nominal delay.

In Table 1, we include results of functional correlation analysis for some ISCAS benchmark circuits. Delays are in ns. We show the delay of the longest path in nominal case ("nominal delay"), from performing iterative topological analysis ("topological long"), and from functional correlation analysis ("functional long"). The percentage shows the extra coupling delay compared to the nominal delay. For C6288, we could not finish the functional analysis as there are too many paths in the circuit, and memory on our machine is not sufficient. As we can see, functional correlation analysis can reduce the pessimism of static timing analysis.

Table 1: Bound of coupling delay

circuit	# of cells	# of pins	Max. level	nominal delay	topological long	functional long
C1355	583	1611	27	14.30	19.06(33%)	17.72(24%)
C1908	407	1574	26	12.15	15.46(27%)	15.37(27%)
C3540	812	3224	36	16.95	19.70(16%)	19.46(15%)
C499	360	1257	20	10.05	13.66(36%)	13.11(30%)
C6288	2435	7187	124	61.7	77.88(26%)	?
C880	311	1008	20	9.25	11.59(25)	11.43(24%)
C2670	971	2788	24	9.95	11.51(16%)	11.51(16%)
C432	190	599	22	11.35	13.75(21%)	13.40(18%)
C5315	1398	5408	30	14.9	16.51(11%)	16.36(10%)
C7552	2056	7474	27	12.45	14.03(13%)	13.78(11%)

In Table 2, we show the number of critical paths and critical gates from different analysis methods as explained in Section 3. Paths which have delays longer than the longest nominal delay in the circuit are considered critical paths. In the column "total path" we show the total number of paths in each circuit. The "topo. long" column gives the number of paths whose coupling delays are larger than the longest nominal delay in the circuit according to the topological analysis. For those paths, we show the number of sensitizable paths in the column "long sensit.". The column "violate" shows the number of long paths after functional correlation analysis. These are the paths whose coupling delays are still larger than the longest nominal delay.

Table 2: Critical paths and gates

circuit	total path	topo. long	long sensit.	violate	neg. slack	topo. gates	funct. gate	run time
C1355	9,276, 432	1,051,730	126,506	55,051	508	449	401	69m
C1908	1,458,112	122,544	68,118	38,139	256	227	218	60
C3540	53,206,636	194,784	46,253	20,039	482	384	355	201m
C499	695,776	64,112	40,528	17,752	284	243	240	20m
C880	16,284	2,248	2,272	1,796	102	66	66	42s
C2670	488,476	5,984	3,276	2,489	262	128	123	7m
C432	483,652	16,738	9,422	2,661	127	119	113	7m
C5315	2,682,418	6,520	1,548	1,425	272	113	107	5m
C7552	1,452,636	5,460	663	590	704	153	146	10m

In many circuits long sensitizable paths are just a small portion of all the total number of paths. The difference between the number of topological long paths ("topo. long") and actual long paths ("violate") may be quite large. The difference between "long sensit." and "violate" shows the effectiveness of detecting path dependent signal correlations in path delay estimation.

The column "neg. slack" gives the number of gates with negative slacks when we only use simple worst delay to compute the required time of each signal. The column "topo. gates" gives the number of critical gates on topological long paths. The column "funct. gates" gives the number of critical gates on long paths after functional correlation analysis. We perform functional correlation analysis on all paths in the circuit, and report run time on Sun Ultra 10.

6. DISCUSSION AND FUTURE WORK

We have applied efficient functional correlation analysis to identify critical paths in a design. This technique is applicable to general circuits. In certain types of circuits, some correlation information can be easily obtained, e.g., the two outputs of a domino gate will be switching in the same direction if they switch during the same phase.

We have discussed several issues in gate resizing to reduce crosstalk delay. It is an attractive technique, as down sizing can also be performed to reduce power if too pessimistic constraints have been used in synthesis. In this paper, we have not addressed the impact of glitches and process variation. We are currently working on these problems.

References

[1] C. J. Alpert, A. Devgan, S.T. Quay, "Buffer Insertion with Accurate Gate and Interconnect Delay Computation", *Proc. 1999 Design Automation Conf.*, pp. 479-84, June 1999.

[2] C. J. Alpert, A. Devgan, S.T. Quay, "Buffer Insertion for Noise and Delay Optimization", *IEEE Trans. Computer-Aided Design*, vol.18, no.11, pp.1633-45, Nov. 1999.

[3] H. C. Chen, D. H. C. Du, and L. R. Liu, "Critical Path Selection for Performance Optimization", *EEE Trans. Comp.-Aided Design*, vol. 12, no. 2, pp.185-195, Feb. 1003.

[4] P. Chen, K. Keutzer, "Towards True Crosstalk Noise Analysis", *Int. Conf. Comp.-Aided Design*, Nov.1999, pp.132-137.

[5] K. T. Cheng, H. C. Chen, "Classification and Identification of Nonrobust Untestable Path Delay Faults", *IEEE Trans. Comp.-Aided Design*, vol. 15, no. 8, pp.845-853, Aug. 1996.

[6] O. Coudert, "Gate Sizing: a General Purpose Optimization Approach", *Proc. European Design & Test Conf.*, pp214-231, March1996.

[7] O. Coudert, R. Haddad, and S. Manne, "New Algorithms for Gate Sizing: A Comparative Study", *Proc. 33rd Design Automation Conf.*, pp734-739, June 1996.

[8] O. Coudert, "Gate Sizing for Constrained Delay/Power/Area Optimization", *IEEE Trans. Very Large Scale Integration System*, vol. 5, no. 4, Dec. 1997.

[9] D. A. Kirkpatrick, A. L. Sangiovanni-Vincentelli, "Digital Sensitivity: Predicting Signal Interaction using Functional Analysis", *Proc. Int. Conf. Comp.-Aided Design*, Nov. 1996, pp. 536-541.

[10] S.S. Sapatnekar, "A timing Model Incorporating the Effect of Crosstalk on Delay and Its Application to Optimal Channel Routing", *IEEE Trans. Computer-Aided Design*, vol.19, no.5, pp.550-9, May 2000.

[11] K. L. Shepard, and V. Narayanan, "Conquering Noise in Deep-Submicron Digital ICs", *IEEE Trans. Design and Test of Computers*, January-March, 1998.

[12] T. Xiao, M. Marek-Sadowska, "Efficient Delay Calculation in Presence of Crosstalk", *IEEE Int. Sym. Quality Electronic Design*, March 2000, pp. 491-497.

[13] T. Xiao, M. Marek-Sadowska, "Worst Delay Estimation in Crosstalk Aware Static Timing Analysis", *Proc. IEEE Int. Conf. Comp. Design*, pp. 115-120, Sept., 2000.

[14] T. Xiao, Chih-Wei Chang, M. Marek-Sadowska, "Efficient Static Timing Analysis in Presence of Crosstalk", *Proc. 13th IEEE Int. ASIC/SOC Conf.*, pp. 335-339, Sept. 2000.