A Method for Reducing the Target Fault List of Crosstalk Faults in Synchronous Sequential Circuits

Hiroshi Takahashi, *Member, IEEE*, Keith J. Keller, Kim T. Le, Kewal K. Saluja, *Fellow, IEEE*, and Yuzo Takamatsu, *Senior Member, IEEE*

***Abstract*—In this paper, we describe a method of identifying a set of target crosstalk faults which may need to be tested in synchronous sequential circuits. Our method classifies the pairs of aggressor and victim lines, using topological and timing information, to deduce a set of target crosstalk faults. In this process, our method also identifies the false crosstalk faults that need not (and/or cannot) be tested in synchronous sequential circuits. Experimental results for ISCAS'89 and ITC'99 benchmark circuits show that the proposed method is CPU time efficient in obtaining the reduced lists of the target crosstalk faults. Also, the lists of the target crosstalk faults obtained by our method are substantially smaller than the sets of all possible combinations of faults.**

***Index Terms*—Crosstalk faults, lists of the target crosstalk faults, synchronous sequential circuits.**

I. Introduction

With the scaling of very large scale integrated (VLSI) feature size and increasing circuit complexity, testing for crosstalk faults has become an important problem. Crosstalk is caused by parasitic coupling between adjacent lines. Crosstalk effects can be divided into two types: crosstalk-induced pulse [1]–[4] and crosstalk-induced delay [2]–[4]. However, as far as crosstalk-induced delays are concerned (the focus of this paper), the existing literature has been limited only to the study of crosstalk-induced delay faults in combinational circuits. Crosstalk-induced delay is introduced due to a simultaneous, or near simultaneous, transitions of an aggressor line and a victim line. If transitions at both lines have the same direction, their transition times are reduced, and the effective delay is reduced (crosstalk delay-speedup). If the transitions at an aggressor line and a victim line have opposite directions, the effective delay is increased (crosstalk delay-slowdown). We refer to these faults as crosstalk-induced transition faults. These unexpected changes in the signal propagation time can cause faulty behavior and it is, therefore, necessary to generate tests for the crosstalk-induced transition faults between aggressor and victim lines. However, the set of all possible combinations of aggressor lines and victim lines is very large and impractical to deal with. For example, in the ISCAS'89 benchmark circuit s38584, the total number of aggressor and victim line pairs is over 400 million. It is clearly important to identify crosstalk-induced transition faults that need to be tested, because certain transition faults may not cause faulty behavior in sequential circuits [9]. Kirkpatrick and his co-authors [7], [8] have proposed using layout information from the physical design to obtain the target crosstalk-fault list and derive tests for these faults only. Additionally, they [7], [8] have proposed the use of static timing analysis to compute delay and develop methods of avoiding associated crosstalk faults during physical design. Similarly, we use the potential correlation between logic design and physical layout to deduce a target fault list for crosstalk faults [12]. However, we believe that our paper [9] is the first research paper in this area that addresses the issue of fault-list reduction of crosstalk-induced transition faults in synchronous sequential circuits. In our method, we identify signal-pairs that should be included in the target crosstalk fault list, and also identify classes of false crosstalk faults that cannot be tested and/or need not be tested. This paper is an expansion of the work presented in [9], and it contains fewer constraints and more experimental results.

Two other methods [10], [11] have been proposed following our work presented at the 2001 International Test Conference [9]. In [10], a crosstalk target identification framework was proposed that is composed of a set of extractors and filters which together identify the target faults. In [11], a capacitive coupling fault model and a crosstalk candidate reduction algorithm is proposed. Both these methods rely heavily on the layout information and are computationally expensive.

Methods for calculating crosstalk noise have also been proposed (e.g., see [14] and [16]). In [16], the authors introduced electrical parameters to calculate the maximum crosstalk noise bound for combinational circuits. In [14], the authors introduced a signal-switching method that is characterized by the set of discontinuous switching windows to reduce the number of violations. These discontinuous switching windows can calculate crosstalk noise more accurately, but the interconnect delay and the gate delay are needed to calculate the number of noise violations.

Manuscript received February 27, 2003; revised August 6, 2003 and January 18, 2004. This research was supported in part by the National Science Foundation under Grant MIP9714034, in part by the Defenses Advanced Research Project Agency (DARPA), in part by the Air Force research Laboratory, in part by Air Force Material Command, in part by USAF under Agreement F30602-00-2-055, in part by the Ministry of Education, Science, Sports, and Culture, Japan, under Grant-in-Aid for Encouragement of Young Scientists 09780293, and in part by the Ministry of Education, Science, Sports and Culture, Japan, under Grant-in-Aid for Scientific Research C(2) (1550043). This paper was recommended by Associate Editor S. Hellebrand.

H. Takahashi and Y. Takamatsu are with the Department of Computer Science, Ehime University, Ehime 790-8577, Japan (e-mail: takahashi@cs.ehime-u.ac.jp).

K. J. Keller is with the Analog Field Applications/Power Management, Texas Instruments, Dallas, TX 75243-4136 USA.

K. T. Le is with the School of Information Sciences and Engineering, University of Canberra, Bruce ACT-2601, Australia.

K. K. Saluja is with the Department of Electrical and Computer Engineering, University of Wisconsin, Madison, WI 53706 USA.

Digital Object Identifier 10.1109/TCAD.2004.837733

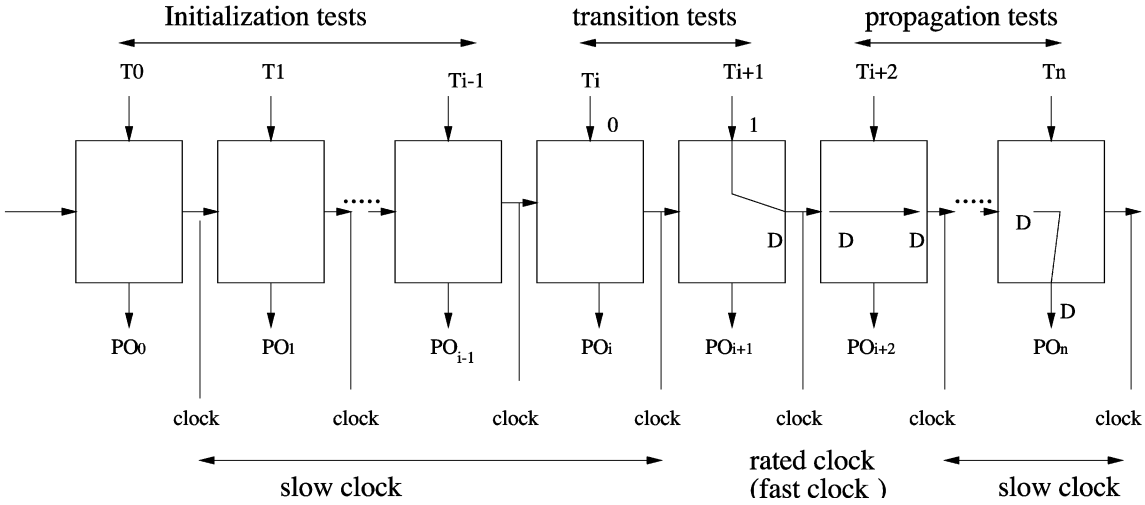

Fig. 1. Testing of a crosstalk-induced transition fault in a sequential circuit.

Since our method, as proposed in this paper, does not need detailed timing and layout information to identify the target faults, our method is much faster. Thus, this method can help reduce the design time and product time-to-market by starting the test generation early in the design cycle. Also, we believe that the method proposed in this paper can be used to identify crosstalk and noise related weaknesses of the design in the early design stages. Finally, the aggressor–victim pairs identified by our method can be provided to placement and routing tools to perform the noise-aware placement and/or routing during physical design.

The four fundamental contributions of this paper are: 1) it demonstrates that the number of all possible crosstalk-induced transition faults in a circuit is very large; 2) it proposes a method to reduce the number of candidate faults using structural and timing information based on the logic-level implementation; 3) it provides a theoretical foundation to identify crosstalk-induced transition faults that cannot occur using idealized assumptions about the operation of a synchronous sequential circuit (this is akin to identifying structurally redundant faults in a circuit); and 4) it demonstrates that the method obtains a reduced list of target crosstalk faults in a relatively small processing time.

This paper is organized as follows. In Section II, we describe the test environment and the theoretical foundation. In Section III, we propose a method for identifying target crosstalk faults and describe an algorithm for obtaining the target fault list. In Section IV, we present experimental results on ISCAS'89 and ITC'99 benchmark circuits to evaluate our method. Finally, we conclude the paper in Section V.

II. Test Environment and Theoretical Foundation

A. Test Environment and Definitions

The test environment that will be used to test crosstalk-induced transition faults in synchronous sequential circuits is as follows. We assume that only a single crosstalk-induced transition fault [2]–[5] can be present in a circuit, because the purpose of our study is to generate a target fault list for test generation. This assumption is akin to the single fault model presently used in literature for testing different classes of faults with ample empirical and theoretical justification.

A crosstalk-induced delay is caused between an aggressor and a victim line. Aggressor lines are denoted as $\mathrm{A} = \{a_1, a_2, \ldots, a_i, \ldots, a_n\}$ and victim lines are denoted as $\mathrm{V} = \{v_1, v_2, \ldots, v_j, \ldots, v_m\}$. A crosstalk-induced delay between an aggressor line and a victim line is denoted as $\mathrm{c}(a_i, v_j)$, where $(a_i, v_j) \in \mathrm{A} \times \mathrm{V} = \{a_i, v_j \,|\, a_i \in A, v_j \in V\}$. A rising (falling) transition at an aggressor line and a victim line are denoted as $a_i \uparrow (\downarrow)$ and $v_j \uparrow (\downarrow)$, respectively. Further, to denote a transition that can be either $\uparrow$ or $\downarrow$, we will use the symbol $\updownarrow$.

We use the slow–fast–slow clock method [6] to test crosstalk-induced transition faults caused by $\mathrm{c}(a_i, v_j)$ in sequential circuits (please refer to Fig. 1). In Fig. 1, the sequential circuit is expressed as an iterative array model. A block represents combinational logic; the left input of a block is the present state with the right output being the next state. The input at the top of each block represents the primary inputs, while the output at the bottom represents primary outputs. In this method, the initialization and propagation tests are applied under slow clocks. During the slow clock, we do not consider the effect of a transition fault. During the transition tests, T_i is applied under the slow clock and T_{i+1} is applied at a rated clock (or fast clock).

Provided the conditions to excite the fault exist (i.e., a necessary signal change occurs at an aggressor line a_i), in this model a crosstalk-induced transition fault on a victim line v_j during a transition test (i.e., time frame T_{i+1}) will excite the fault.

Further, the effect of a crosstalk-induced transition fault may appear at a PO_{i+1}, or may be captured at one or more of the next state output(s) of the block and will have to be propagated using slow clocks. The above discussion, as well as the study of crosstalk-induced transition faults in combinational circuits reported in [2]–[5], leads to the following necessary conditions to test crosstalk-induced transition faults in sequential circuits.

Condition 1: Excitation of a Crosstalk-Induced Transition Fault [2]–[5]: It has been demonstrated in [2]–[5] that $c(a_i, v_j)$ are induced only if $a_i \updownarrow$ and $v_j \updownarrow$ occur at appropriate times. Let us consider the time frame T_{i+1}. Let the transition time at an

aggressor line be $a_i(t)$, and let the transition time at the victim line be $v_j(t)$. A $c(a_i, v_j)$ is induced at the line v_j provided

$$(v_j(t) - \Delta \text{ unit delays}) \leq a_i(t) \leq (v_j(t) + \Delta \text{ unit delays})$$

where Δ is 1 or more unit delays and $[v_j(t) - \Delta$ unit delays, $v_j(t) + \Delta$ unit delays$]$ is called the Δ timing window. The transitions must be limited to this window to cause slowdown or speedup on the line v_j. From experimental results, the authors in [2]–[5] demonstrated that v_j and a_i change must occur within one or two unit delays to excite slowdown or speedup effect. Further, if $a_i \updownarrow$ and $v_j \updownarrow$ have the same direction (i.e., both $\uparrow$ or both $\downarrow$), then the effective delay is said to be reduced (crosstalk delay-speedup) and if $a_i \updownarrow$ and $v_j \updownarrow$ have the opposite directions, the effective delay is said to be increased (crosstalk delay-slowdown).

Condition 2: Capturing the Crosstalk-Induced Transition Fault: If a $v_j \updownarrow$, which is affected by the crosstalk-induced delay, arrives at the fanin of a flip-flop after the normal clock-edge, then only an incorrect value can be captured in the flip-flop.

Condition 3: Observing the Crosstalk-Induced Transition Fault at a Primary Output (PO): If the incorrect value caused by the crosstalk-induced transition fault propagates to at least one primary output, then the incorrect value is observed at the primary output.

Condition 1 must be satisfied for a crosstalk-induced transition fault to be excited. The resulting fault effect may appear at a PO in the same time frame in which case the transition fault is detected. Alternatively, both conditions 1 and 2 must be satisfied when one or more incorrect values are captured in a flip-flop (or more than one flip-flop), denoted as D in Fig. 1 at timeframe T_{i+1}. In this case, Condition 3 must also be satisfied to propagate the fault effect to a PO of a later time frame. Thus, for the generation of fault lists, we need to focus on the time frame T_{i+1}.

The test environment shown in Fig. 1 may appear similar to that of testing for delay faults. However, there are important differences while testing for crosstalk-induced transition faults; these differences primarily lie in the conditions for exciting the crosstalk-induced transition faults. Therefore, in our method for fault list reduction, we focus on checking the conditions required to excite a crosstalk-induced transition fault between an aggressor and victim line.

Additionally, to help reduce the fault list further, conditions that are required for propagation of a transition fault along a path can be borrowed from conventional test generators; however, these methods are not used in this paper.

Definition 1: A crosstalk-induced transition fault is a *false crosstalk fault* if it is unexcitable or if it need not, and/or cannot, be tested.

For the purpose of this paper, unexcitable faults are those faults which do not satisfy Condition 1. For certain excitable faults, the transition fault can be excited and propagated to a fanin of a flip-flop or a primary output, but the incorrect value cannot be captured in a flip-flop, nor can it be observed at a primary output when a normal clock-edge occurs. We refer to these transition faults as the faults that need not be tested.

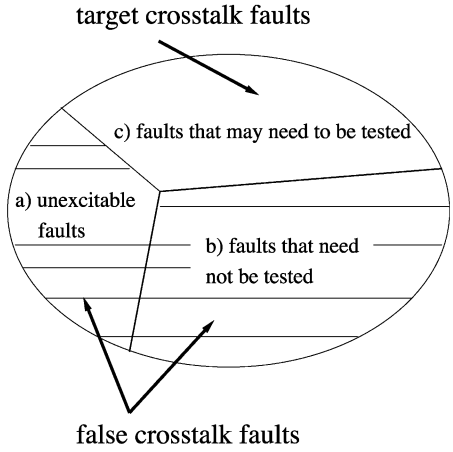

Fig. 2. Classification of crosstalk-induced transition faults.

Definition 2: A crosstalk-induced transition fault that may need to be tested is called a *target crosstalk fault*.

Fig. 2 shows the classification of the crosstalk-induced transition faults in sequential circuits. The false crosstalk faults are faults that belong to sets a and b in Fig. 2. The target crosstalk faults are faults that belong to set c in Fig. 2.

B. Theoretical Foundation

Before providing the theory and methods for fault-list reduction, we state our assumptions.

Assumption 1: We assume that clock-skew is not present, and that all flip-flops are positive edge triggered flip-flops.

Assumption 2: The clock cycle of the circuit is determined by the delay of the structurally longest path in the circuit.

Assumption 3: A crosstalk-induced delay on a v_j is excited if the $a_i \updownarrow$ occurs within the Δ timing window of a $v_j \updownarrow$ [2]–[5].

Assumption 4: We use the unit delay model, where each gate is assumed to have the same delay. This model specifies the interval between the time at which the inputs to the gate change and the time the output changes. No delay is assumed for interconnects [15].

Assumption 5: The size of extra delay caused by the crosstalk-induced delay is one unit delay.

Assumption 6: The sum of the setup and hold times of the flip-flop is one unit delay.

Definition 3: An edge of the clock on which the flip-flops do not latch data is called the *ineffective clock edge*.

Note that for a positive (negative) edge trigger design, the falling (rising) edge is the ineffective clock edge. However, the studies carried out in this paper correctly account for the faults that may be caused by the ineffective clock edge.

The candidate v_j and a_i are the clock-lines, primary inputs, and gate outputs. For simplicity, we refer to the primary inputs and the gate outputs as "lines" and clock signals as "clock-lines". The interaction between a_i and v_j are divided into the following four cases and these cases are also shown graphically in Fig. 3.

Case 1) a_i and v_j are lines.
Case 2) a_i is a line and v_j is a clock-line.
Case 3) a_i is a clock-line and v_j is a line.
Case 4) a_i and v_j are clock-lines.

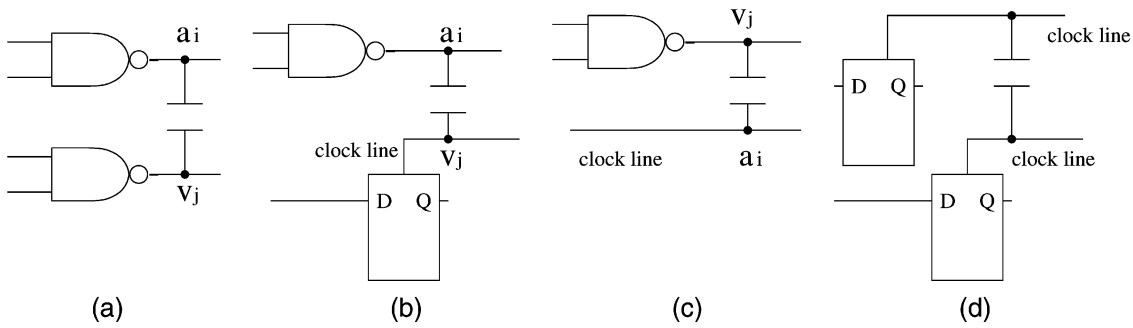

Fig. 3. Examples of four cases for crosstalk-induced transition faults. (a) Example of Case 1. (b) Example of Case 2. (c) Example of Case 3. (d) Example of Case 4.

Next, we describe the target crosstalk faults and the false crosstalk faults for each case listed above. We will describe all the formal and theoretical results assuming the victim lines are lines that are included in the longest (critical) path. However, this condition is relaxed in the algorithm implementation; therefore, the experimental results in Section IV are presented even when the victim line is not on the longest path.

Case 1) a_i **and** v_j **are lines**.

Lemma 1: If a $v_j \updownarrow$ is such that the crosstalk-induced transition fault is caused by crosstalk delay-speedup, then the incorrect value cannot be captured at any flip-flop or any primary output by the normal clock-edge.

Lemma 2: If a transition that propagates along a longest path is affected by the crosstalk delay-slowdown, then the incorrect value may be captured at a flip-flop or a primary output by the normal clock-edge.

Lemma 3: If a v_j is not included in any longest path, the transition that is affected by crosstalk delay-slowdown must arrive at a fanin of a flip-flop or a primary output before the correct clock-edge.

From Lemmas 1 and 2, the following theorem holds for crosstalk-induced transition faults that may need to be tested in sequential circuits.

Theorem 1: For every v_j, which is included in a longest path, the crosstalk-induced transition fault caused by crosstalk delay-slowdown between the v_j and all other lines in the circuit are target crosstalk faults in a sequential circuit.

From Lemma 3, we can deduce the following theorem to identify the crosstalk-induced transition faults that need not be tested in sequential circuits.

Theorem 2: For every a_i in the circuit and every v_j which is not included in any longest path, the crosstalk-induced transition fault caused by $c(a_i, v_j)$ is a false crosstalk fault.

Case 2) a_i **is a line and** v_j **is a clock-line.**

Lemma 4: If a_i is a line and v_j is a clock-line, then this $c(a_i, v_j)$ cannot satisfy the Condition 1 for exciting the crosstalk-induced transition fault because a_i would cause a delay on the ineffective clock edge with no degradation to the circuit performance.

Lemma 4 leads to the following theorem, which identifies the crosstalk-induced transition faults that cannot be tested in sequential circuits.

Theorem 3: If a_i is the line and v_j is the clock-line, the crosstalk-induced transition fault caused by $c(a_i, v_j)$ in Case 2 is a false crosstalk fault which is an unexcitable fault in a sequential circuit.

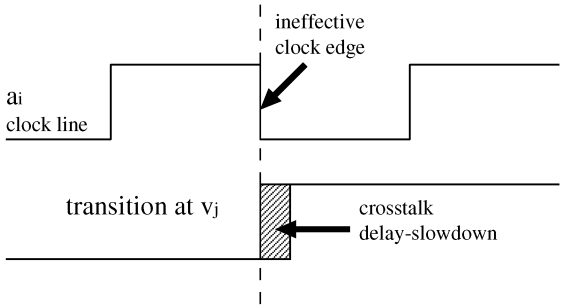

Fig. 4. Explanation of Lemma 5.

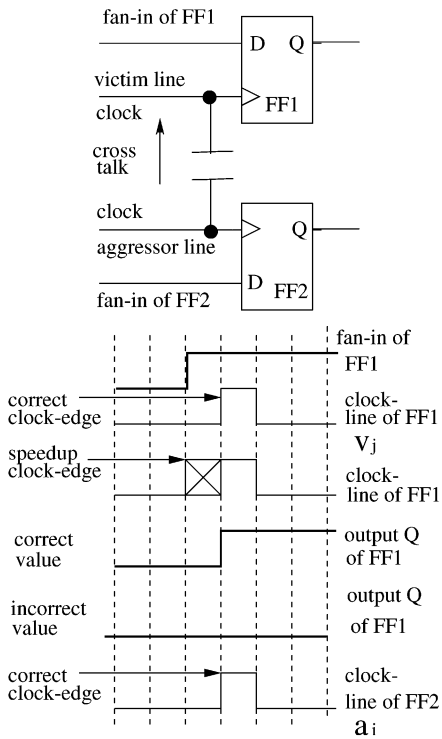

Fig. 5. Excitation and capture of crosstalk-induced speedup at clock line.

Case 3) a_i **is a clock-line and** v_j **is a line.**

Lemma 5: The ineffective clock edge can be an aggressor to a gate line included in the longest path.

Fig. 4 is used to explain Lemma 5. In Fig. 4, a victim line is a line that is included in the longest path and an aggressor line is

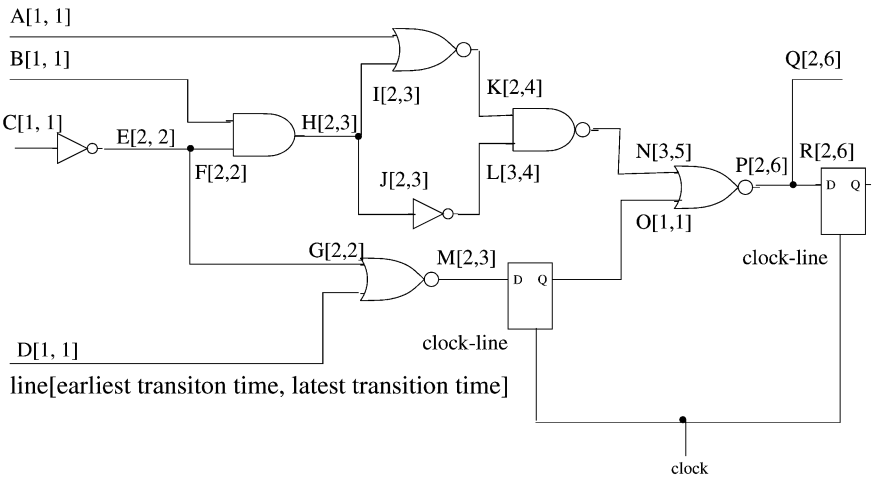

Fig. 6. Example of the transition times of each line.

a clock line. If a rising transition at a victim line and the ineffective clock edge occur simultaneously, or near simultaneously, a crosstalk delay-slowdown may occur at the victim line. Lemma 5 in this paper covers this situation.

From Lemma 5, we can deduce the following theorem to identify the crosstalk-induced transition faults that may need to be tested in sequential circuits.

Theorem 4: For every v_j which is included in the longest path, the transition fault caused by crosstalk delay-slowdown between the v_j and the ineffective edge of the aggressor clock lines are target crosstalk faults in a sequential circuit.

Case 4) a_i and v_j are the clock-lines.

Lemma 6: If a transition is propagated along a longest path that arrives at a fanin of a flip-flop, then the incorrect value is captured by the clock-edge affected by crosstalk delay-speedup.

From Lemma 6, the following theorem holds on the crosstalk-induced transition faults in sequential circuits.

Theorem 5: The crosstalk-induced transition fault caused by crosstalk delay-speedup between a clock-line and a victim clock-line of a flip-flop whose fanin is included in a longest path is a target crosstalk fault.

Fig. 5 shows an example of the excitation and the capturing of a crosstalk-induced transition fault at the clock-line of FF1. In this example, clock-edges at clock-lines of FF1 and FF2 have the same direction, and the crosstalk delay-speedup is excited. A fanin of FF1 is included in a longest path. The transition at the fanin of FF1 does not arrive before the speedup clock arrives at FF1 because the transition of fanin of FF1 is propagated along longest path. Therefore, FF1 captures the faulty value 0. Note that we do provide the conditions for checking whether or not a fault effect that is captured in a flip-flop propagates to a primary output. Therefore, the target fault list may include the crosstalk-induced transition faults that do not satisfy Condition 3 of Section II-A.

III. Identification of Target Crosstalk Faults

Based on the theorems in Section II-B, in this section we derive the conditions for the target crosstalk faults that need to be tested. If a crosstalk-induced transition fault caused by $c(a_i, v_j)$ does not satisfy the condition of a target fault, then we determine the crosstalk-induced transition fault as a false crosstalk fault. We then propose a method for checking whether or not the crosstalk-induced transition fault caused by $c(a_i, v_j)$ satisfies the conditions of the target crosstalk faults.

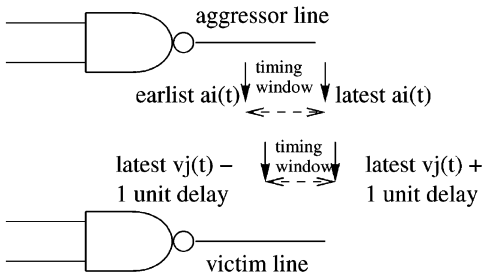

Fig. 7. Timing windows of a_i and v_j.

From the discussion in Section II-B, we consider the conditions for the target crosstalk faults for Cases 1, 3, and 4 as well as the condition for the false crosstalk faults for Case 2; these are given below.

Case 1) a_i and v_j are lines. A crosstalk-induced transition fault caused by $c(a_i, v_j)$ is a target crosstalk fault if the following conditions are satisfied.

C1) v_j must be included in a longest path.

C2) The relationship between $a_i(t)$ and $v_j(t)$ satisfies the following inequality:

$$(v_j(t) - \Delta \text{ unit delays}) \leq a_i(t) \leq (v_j(t) + \Delta \text{ unit delays}).$$

Case 2) a_i is a line and v_j is a clock-line: $c(a_i, v_j)$ cannot satisfy the condition for exciting a crosstalk-induced transition fault because a_i would cause a delay on the ineffective clock edge with no degradation to the circuit performance.

Case 3) a_i is a clock-line and v_j is a line. A crosstalk-induced transition fault caused by $c(a_i, v_j)$ is a target crosstalk fault if the following conditions are satisfied.

C3) v_j must be included in a longest path.

C4) The relationship between $a_i(t)$ and $v_j(t)$ satisfies the following inequality:

$$(v_j(t) - \Delta \text{ unit delays}) \leq a_i(t) \leq (v_j(t) + \Delta \text{ unit delays}).$$

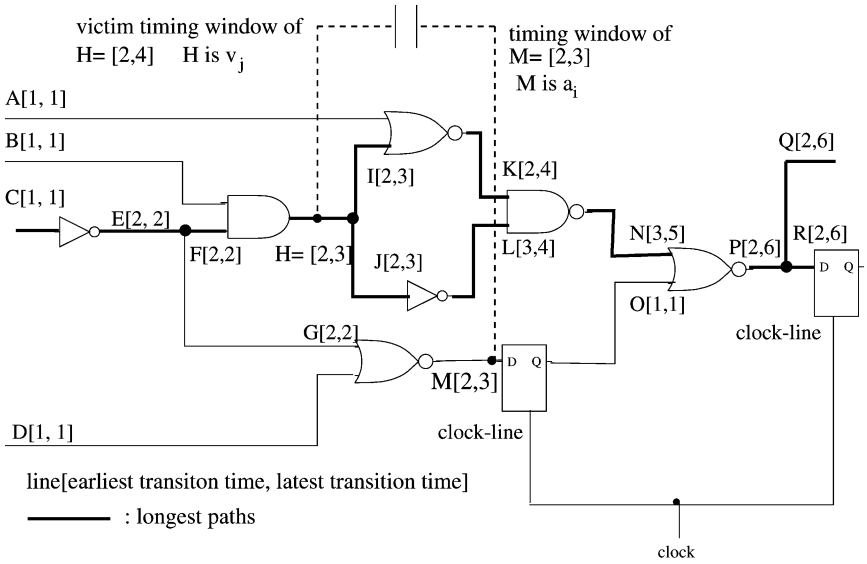

Fig. 8. Example of the target crosstalk fault of Case 1.

Case 4) a_i and v_j are clock-lines. A crosstalk-induced transition fault caused by $c(a_i, v_j)$ is a target crosstalk fault if the following condition is satisfied.

C5) The fanin of the flip-flop with the victim clock-line is included in a longest path.

Described next in our method, we consider both the topological and the timing information. For timing information, we calculate the latest transition time and the earliest transition time at each line. We then find the lines included in longest paths by using the topological and timing information.

In this paper, we use the unit delay model as we do not have accurate delay information. Under the unit delay model, all gate delays are the same irrespective of their type and input condition. If accurate timing information is available, we believe that the proposed method can easily take into account this delay information.

The latest and earliest transition times are defined as follows.

Definition 4: The *latest transition time* at a line is the maximum delay on any path from any primary input or the output of a flip-flop to this line.

The latest transition time is recursively calculated for a line L using the following equations.

1) Line L is a primary input:

$$\text{latest transition time for L} = 1 \text{ unit delay.}$$

2) L is an output of gate G having n inputs $\mathrm{I}_1, \mathrm{I}_2, \ldots, \mathrm{I}_n$:

$$\text{latest transition time for L} = \mathrm{MAX}\{\text{latest transition time for } \mathrm{I}_j, j = 1, \ldots, n\} + \text{delay of gate G.}$$

3) Line L is a fanout branch:

$$\text{latest transition time for L} = \text{latest transition time for the fanout stem.}$$

4) Line L is an output of a flip-flop:

$$\text{latest transition time of L} = 1 \text{ unit delay.}$$

Definition 5: The *earliest transition time* at a line is the minimum delay on any path from any primary input or the output of a flip-flop to this line.

The earliest transition time is recursively calculated for a line L using the following equations.

1) Line L is a primary input:

$$\text{earliest transition time for L} = 1 \text{ unit delay.}$$

2) L is an output of gate G having n inputs $\mathrm{I}_1, \mathrm{I}_2, \ldots, \mathrm{I}_n$:

$$\text{earliest transition time for L} = \mathrm{MIN}\{\text{earliest transition time for } \mathrm{I}_j, j = 1, \ldots n\} + \text{delay of gate G.}$$

3) Line L is a fanout branch:

$$\text{earliest transition time for L} = \text{earliest transition time for the fanout stem.}$$

4) Line L is an output of a flip-flop:

$$\text{earliest transition time for L} = 1 \text{ unit delay.}$$

Using the rules given above, the earliest and latest transition times are calculated for each line. An example of the earliest and latest transition times for all lines in a circuit is shown in Fig. 6; this example assumes that all gates have unit delay. In our study, the aggressor line has a timing window of [earliest $a_i(t)$, latest $a_i(t)$]. Similarly, the victim line also has a timing window of [(latest $v_j(t) - \Delta$ unit delays), (latest $v_j(t) + \Delta$ unit delays)], and the value of Δ between the a_i and the v_j is assumed to be one or two unit delays. This assumption is justified by the experimental results reported in [2]–[5].

In order to identify the target faults, we check whether the timing window at a_i overlaps with the timing window at v_j. Fig. 7 shows an example of timing windows of an aggressor and victim line.

In [14], signal switching that is characterized by the set of discontinuous switching windows is introduced to calculate a more accurate effect of crosstalk-induced delay. The delay model in [14] consists of the interconnect and gate delays.

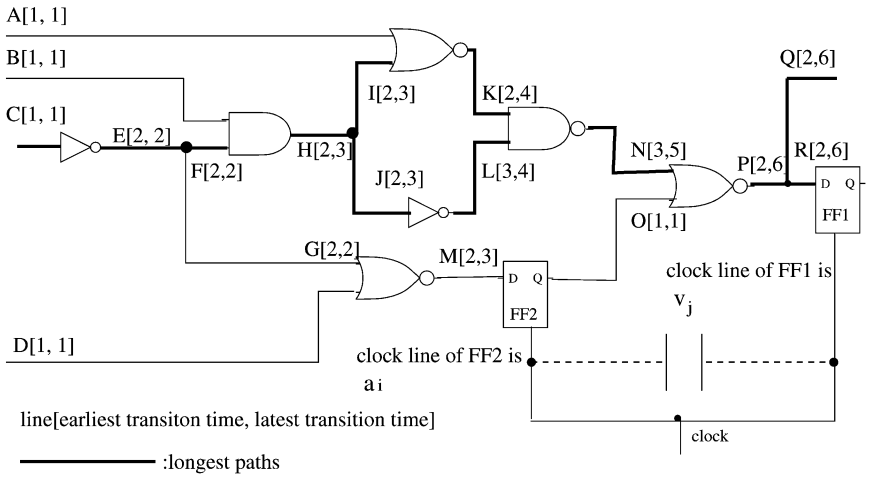

Fig. 9. Example of the target crosstalk fault of Case 4.

TABLE II
ITC BENCHMARK CIRCUIT GENERAL INFORMATION

circuit names	All possible pairs	No. of lines in LPs	No. of candidate pairs
b01	1,892	11	629
b02	702	15	510
b03	20,880	42	9,396
b04s	346,332	40	27,534
b06	2,862	10	682
b07s	169,332	36	17,940
b08	26,406	21	4,209
b09	24,806	64	13,320
b10	33,306	23	5,174
b11s	225,150	59	30,300
b12	1.060M	46	62,221
b13s	107,912	17	7,620
b14s	22.28M	184	918,525
b15s	77.84M	216	2.030M
b17s	580.6M	705	18.21M
b20s	88.29M	178	1.770M
b21s	95.66M	178	1.838M
b22s	226.5M	180	2.857M
LP: Longest Path			
M $=10^6$			

TABLE I
ISCAS BENCHMARK CIRCUIT GENERAL INFORMATION

circuit names	All possible pairs	No. of lines in LPs	No. of candidate pairs
s27	272	9	193
s208	13,110	18	2,196
s298	18,360	10	1,639
s344	33,672	21	4,356
s349	34,040	21	4,378
s382	32,942	29	6,666
s386	29,412	58	10,638
s400	34,410	30	7,004
s420	53,130	31	7,626
s444	41,820	38	9,450
s526	46,872	10	2,607
s526n	47,306	10	2,618
s641	187,056	80	36,118
s713	199,362	84	39,098
s820	97,032	43	14,250
s832	95,790	43	14,160
s838	208,392	57	27,816
s953	193,160	20	10,296
s1196	314,160	73	42,266
s1238	291,060	45	25,101
s1423	558,756	65	54,260
s5378	8.955M	70	221,970
s9234	34.15M	370	2.258M
s13207	74.83M	165	1.547M
s15850	107.8M	341	3.766M
s35932	317.8M	9,734	196.0M
s38417	568.5M	74	1.911M
s38584	429.2M	182	4.057M
LP: Longest Path			
M $=10^6$			

The method proposed in this paper can easily incorporate discontinuous switching windows provided the accurate delay information is available.

The proposed method consists of two phases, and is relatively fast in obtaining the lists of target faults because the method uses structure and timing information based on the logic-level implementation. Procedures for implementing Phases 1 and 2 are described below.

[Procedures for identifying target crosstalk faults]

Phase 1:

Step 1) Calculate the earliest and latest transition times at each line.

Step 2) Identify the longest paths in the circuit using the maximum value of the latest transition time.

Step 3) Identify the lines that are included in the longest paths by using the latest transition time and the topological information. Add these lines to the set of victim lines (Set_V).

Phase 2:

Step 1) Select a v_j from the Set_V and remove from Set_V.

Step 2) Compare the timing window of the selected v_j, as well as the earliest and the latest transition times of other gate outputs. If the timing window [earliest $a_i(t)$, latest $a_i(t)$] of a_i overlaps with the timing window [(latest $v_j(t)$ − 1 unit delay), (latest $v_j(t)$ + 1 unit delay)] of v_j, then add the crosstalk-induced transition fault caused by the selected $c(a_i, v_j)$ to the list of the target crosstalk faults.

TABLE III
ISCAS EXPERIMENTAL RESULTS FOR IDENTIFYING TARGET FAULTS (Δ = 1 UNIT DELAY)

circuit names	No. of target faults (case 1)	No. of target faults (case 3)	No. of target faults (case 4)	target faults (%)	No. of false faults (case 1)	No. of false faults (case 2)	No. of false faults (case 3)	run times (sec.)	Improvement of the method (%)
s27	59	4	2	34	85	17	26	0.3	42
s208	561	4	0	26	1,491	0	140	1.1	10
s298	501	3	13	32	849	136	137	1.3	4
s344	857	3	14	20	2,986	184	312	1.6	7
s349	863	3	14	20	3,001	185	312	1.7	7
s382	1,667	3	80	26	3,582	728	606	2.1	10
s386	4,197	14	10	40	5,721	344	352	2.1	28
s400	1,754	3	80	26	3,796	744	627	2.2	10
s420	1,068	3	0	14	6,062	0	493	2.0	6
s444	2,066	6	80	23	5,686	820	792	2.3	11
s526	853	3	20	34	1,307	217	207	2.2	3
s526n	856	3	20	34	1,314	218	207	2.3	3
s641	2,807	3	0	8	31,753	0	1,555	4.1	9
s713	3,045	3	0	8	34,419	0	1,631	4.3	10
s820	7,597	7	8	53	5,776	624	238	3.0	11
s832	7,612	7	8	54	5,675	620	238	2.9	11
s838	2,122	3	0	8	23,870	0	1,821	3.9	3
s953	2,238	3	56	22	6,542	880	577	4.0	3
s1196	7,316	6	0	17	33,564	0	1,380	4.6	6
s1238	3,555	3	0	14	20,700	0	843	4.3	3
s1423	5,066	3	73	9	43,489	748	4,881	5.8	3
s5378	53,250	3	0	24	156,190	0	12,527	22.7	2
s9234	347,394	12	454	15	1.815M	11,688	84,348	43.5	5
s13207	106,323	9	668	7	1.321M	8,651	110,376	61.3	6
s15850	101,314	32	1,192	3	3.439M	20,766	203,545	75.8	1
s35932	21.61M	864	497,376	11	151.9M	5.134M	16.82M	222.1	23
s38417	123,263	3	1,635	7	1.641M	23,843	121,061	195.0	0.1
s38584	316,185	7	1,451	8	3.454M	20,717	264,257	163.9	0.3
LP: Longest Path									
M $=10^6$									

Step 3) If the $\mathrm{Set\_}V$ is not empty, then go to Step 1.
Step 4) Identify the clock-line of each flip-flop whose fanin line is included in the longest path(s). These clock-lines belong to the set of V-lines ($\mathrm{Set\_}V$).
Step 5) Add the pair (v_j and other clock-line) to the list of the target crosstalk faults.

Example: We illustrate the proposed method by using the sequential circuit shown in Figs. 8–9 for the following scenarios.

1) By using Phase 1, we generate $\mathrm{Set\_}V$. We identify the longest paths in the circuit. Add gate outputs C, E, H, K, L, N, and P in Fig. 8, which are included in the longest paths, to the $\mathrm{Set\_}V$.
2) In Phase 2, we select the line H as a v_j from the $\mathrm{Set\_}V$. Note that the original calculated timing window of H is $[2,3]$ (equal to both I and J), but the victim timing window is $[2,4]$ (from $[(\text{latest } v_j(t) - 1 \text{ unit delay}), (\text{latest } v_j(t) + 1 \text{ unit delay})]$).
3) We now check the crosstalk-induced transition fault caused by $c(M,H)$ as shown in Fig. 8. Because the timing window $[\text{earliest } M(t) = 2, \text{latest } M(t) = 3]$ at line M (aggressor line) overlaps with the timing window $[(\text{latest } H(t) - 1 \text{ unit delay}) = 2, (\text{latest } H(t) + 1 \text{ unit delay}) = 4]$ at line H (victim line), we add the crosstalk-induced transition fault caused by $c(M,H)$ to the list of the target crosstalk faults.
4) In Phase 2, we consider line N as a v_j from the $\mathrm{Set\_}V$.
5) We check the crosstalk-induced transition fault caused by $c(B,N)$ (See Fig. 8). Because the timing window $[\text{earliest } B(t) = 1, \text{latest } B(t) = 1]$ at line B (aggressor line) does not overlap with the timing window $[(\text{latest } N(t) - 1 \text{ unit delay}) = 4, (\text{latest } N(t) + 1 \text{ unit delay}) = 6]$ at line N (victim line), this pair is a false crosstalk fault. Therefore, the crosstalk-induced transition fault caused by $c(B,N)$ is not added to the list of target crosstalk faults.
6) In Phase 2, the pair of clock lines of FF1 and FF2 is added to the list of the target crosstalk faults, because a fanin line R of FF1 is included in the longest path. (See Fig. 9.)

IV. EXPERIMENTAL RESULTS

We implemented the procedure for identifying target crosstalk faults in sequential circuits in C code to run on Sun Ultra 10 workstation with 440-MHz CPU. We conducted the following experiments on ISCAS'89 and ITC'99 benchmark circuits.

Experimental results for ISCAS circuits are shown in Tables I, III, IV, and V. Our experiments include results for $\Delta = 1$ unit delay and $\Delta = 2$ unit delay timing windows of the victim line. Note that the value of Δ is expected to be one or two units, as discussed in this paper earlier, and as observed by the researchers in [2]–[5]. Further, in this paper we also considered "almost longest paths" to estimate the growth of the number of

TABLE IV
ISCAS EXPERIMENTAL RESULTS FOR IDENTIFYING TARGET FAULTS (Δ = 2 UNIT DELAYS)

circuit names	No. of target faults (case 1)	No. of target faults (case 3)	No. of target faults (case 4)	target faults (%)	No. of false faults (case 1)	No. of false faults (case 2)	No. of false faults (case 3)	Improvement of the method (%)
s27	84	6	2	48	60	17	24	43
s208	835	7	0	38	1,217	0	137	10
s298	713	5	13	45	637	136	135	5
s344	1,274	5	14	30	2,569	184	310	7
s349	1,282	5	14	30	2,582	185	310	7
s382	2,474	8	80	38	2,775	728	601	11
s386	6,009	24	10	57	3,909	344	342	29
s400	2,604	8	80	38	2,946	744	622	11
s420	1,631	5	0	21	5,499	0	491	6
s444	3,191	11	80	35	4,561	820	787	12
s526	1,183	5	20	46	977	217	205	3
s526n	1,191	5	20	46	979	218	205	3
s641	4,490	5	0	12	30,070	0	1,533	10
s713	4,898	5	0	13	32,566	0	1,629	10
s820	9,730	12	8	69	3,643	624	233	12
s832	9,700	12	8	69	3,587	620	233	12
s838	3,296	5	0	12	22,696	0	1,819	4
s953	3,448	5	56	34	5,332	880	575	3
s1196	11,054	10	0	26	29,826	0	1,376	6
s1238	5,568	6	0	22	18,687	0	840	3
s1423	7,635	5	73	14	40,920	748	4,879	4
s5378	81,688	7	0	37	127,752	0	12,523	2
s9234	551,528	18	454	24	1.610M	11,688	84,342	5
s13207	170,181	17	668	11	1.257M	8,651	110,368	1
s15850	164,209	48	1,192	4	3.376M	20,766	203,529	1
s35932	34.62M	1,440	497,376	18	138.9M	5.134M	16.82M	25
s38417	195,321	5	1,635	10	1.569M	23,843	121,059	0.1
s38584	478,848	11	1,451	12	3.291M	20,717	264,253	0.3

LP: Longest Path
M = 10^6

target faults that belong to Cases 1, 3, and 4. Similar results are given for ITC benchmark circuits in Tables II and VI.

A. Experimental Results for ISCAS'89

In Table I, we give general information about the ISCAS'89 circuits. The column "all possible pairs" contains all possible combinations of pairs of gate outputs, primary inputs and clock-lines of the flip-flops. The column "number of lines in LPs" contains the number of lines included in all longest paths in the circuit. This number shows the size of $\mathrm{Set\_}V$ obtained by Phase 1. The column "number of candidate pairs" contains the total number of combinations between lines included in the longest paths and all other lines in the circuit. In this paper, we consider all combination pairs of gate outputs, primary inputs and clock-lines of the flip-flops, because we do not use any layout information to reduce the number of combinations of signal-pairs as potential candidate pairs.

Table III shows the experimental results for a timing window Δ of one unit delay. The columns "no. of target faults (Case 1)," "no. of target faults (Case 3)," and "no. of target faults (Case 4)" contain the total number of target crosstalk faults for these cases, respectively, that are obtained by our method. These numbers are obtained during Phase 2. The column "target faults (%)" contains a ratio of the total number of target crosstalk faults for Cases 1, 3, and 4 to the total number of candidate pairs. The columns "no. of false faults (Case 1)," "no. of false faults (Case 2)," and "no. of false faults (Case 3)" contain the total number of false crosstalk faults for these cases that are obtained by the theory proposed in Section II-B.

The column "run times" contains the CPU times consumed by the proposed method. The column "improvement of the method (%)" is defined as (the number of target faults obtained by the proposed method) $\div$ (the number of pairs that satisfy the excitation condition between an aggressor line and a victim line—these pairs are identified by using only the timing information) $\times$ 100.

Information from the values in the column "target faults" and the column "improvement of the method" shows that the method is useful in reducing the number of target crosstalk faults, while considering the longest paths information. Information from the values in the column "run times" shows that the speed of the proposed method is fast and the CPU time of the largest circuit is a few minutes. In these experiments, we made no effort to optimize runtime. We believe that runtime can be further reduced with optimization of implementing the tool.

As an example, let us consider the s38584 circuit in Tables I and III. The number of all combination pairs of the gate outputs, primary inputs and clock-lines of the flip-flops is 429 173 372. By using Phase 1 we identify 182 lines that are included in the longest paths of the circuit. The total number of candidate pairs reduces to 4 056 744. As a result of Phase 2, using the timing information, 316 185 faults are identified to belong to the list

TABLE V
COMPARISON OF NUMBER OF TARGET FAULTS FOR VARYING SIZES OF ALMOST LONGEST PATHS

	Number of target faults belonging to Cases 1, 3 or 4				
circuits	LP	[LP-1, LP]	[LP-2, LP]	[LP-3, LP]	[LP-4, LP]
s27	65	67	67	67	119
s208	565	565	565	1121	1121
s298	517	673	673	1218	2280
s344	874	874	926	1004	1004
s349	880	932	1,010	1,010	1,010
s382	1,750	3,748	6,518	6,518	9,287
s386	4,221	4,989	8,729	8,729	8,729
s400	1,837	3,887	6,790	6,790	9,739
s420	1,071	1,071	1,071	1,071	1,071
s444	2,152	2,152	3,740	6,040	8,695
s526	876	1,080	1,080	6,457	9,982
s526	879	1,083	1,083	6,990	10,527
s641	2,810	2,810	2,810	2,940	3,064
s713	3,048	3,048	3,048	3,048	3,345
s820	7,612	7,612	12,856	12,856	20,388
s832	7,627	7,627	12,853	12,853	20,376
s838	2,125	2,125	2,125	2,125	2,125
s953	2,297	5,457	5,508	9,411	11,858
s1196	7,322	7,544	7,544	8,195	8,297
s1238	3,558	7,363	7,363	8,049	8,205
s1423	5,142	5,255	5,372	5,484	5,484
s5378	53,253	53,253	104,220	104,530	168,074
s9234	347,860	347,860	347,860	347,860	347,860
s13207	107,000	107,000	107,000	107,830	107,830
s15850	102,538	102,538	102,538	102,538	102,538
s35932	22.11M	22.11M	22.11M	22.11M	22.11M
s38417	124,901	124,901	124,901	124,901	124,901
s38584	317,643	317,643	317,643	317,643	317,643
LP: Longest Path					
$M = 10^6$					

of the target crosstalk faults in Case 1, 7 faults in Case 3, and 1 451 faults in Case 4. Thus 3 454 127 faults in Case 1 are false crosstalk faults, 20 717 faults in Case 2, and 264 257 faults in Case 3 are false crosstalk faults.

By using the proposed method, we identify target crosstalk faults for the s38584 circuit in 163.9 s. The percentage of target crosstalk faults for the s38584 circuit is 8% of the total number of candidate pairs. If we do not consider longest path information (i.e., only using the timing information), then 104 393 675 pairs are identified as the target crosstalk faults by Phase 2. From the value of the improvement of the method in Table III, the number of target faults obtained by the method while considering longest path information is about a thousandth of the number of target faults obtained by the method without considering longest path information.

We note that the number of target faults for s15850, s35932, and s38584 circuits, which have very large number of candidate pairs, reduces substantially by using our method.

We also compare the results of Table III with those of Table IV to estimate the effect of varying the size of Δ in the timing window. In addition, we compare the results for the timing window with Δ of one unit delay against those for the timing window with Δ of two unit delays to estimate the effect of varying the size of Δ in the timing window. The target crosstalk fault set obtained by the timing window with Δ of two unit delays is about 1.5 times that of the target crosstalk fault set obtained by the timing window with Δ of one unit delay.

To study the impact of crosstalk-induced delays consisting of more than one unit delay on the size of the target fault list, we conducted an experiment by defining an "almost longest path" to be a path which is shorter than the longest path by a specified value. Thus, for measuring the impact of an induced delay, which is of duration k units, all we need to do is obtain the initial $\mathrm{Set\_}V$ in Phase 1 of our method as the set of lines in all paths that are of length $\mathrm{LP} - \mathrm{k} + 1$ or more (where LP is the length of the longest path in the circuit). The results of this experiment are summarized in Table V. This table shows the growth in the number of target faults at different lengths of the almost longest paths. The column "[LP-1, LP]" contains the number of the target crosstalk faults that are obtained by this experiment. In Phase 1, we obtained the set of victim lines included in the paths having length LP-δ, where $\delta = 1 \ldots \mathrm{E}$. In Phase 2, we use the size of Δ timing window to be one unit delay to identify the target crosstalk fault list. From the results of Table V, it is evident that even if the length of the almost longest path is varied the growth in the number of target crosstalk faults in ISCAS'89 benchmark circuits is not very large. As for the case of the longest path, for all other cases, too, the total number of target crosstalk faults is substantially smaller and only a fraction of all possible line pairs.

B. Experimental Results for ITC'99

Table II gives the same general testing information for the ITC'99 benchmark circuits as Table I does for the ISCAS'89 benchmark circuits.

Reduction results are shown for ITC benchmark circuits for a Δ of one unit delay in Table VI. Results are very similar to those of the ISCAS'89 circuits.

C. Practical Testing and Experimental Runtimes

Finally, we must add that, as also observed by Chen *et al.* [13] in their experimental study, the number of crosstalk faults that can indeed be tested is expected to be small. Chen *et al.* [13] in their study of a number of circuits considered only 100 fault sites as candidates for crosstalk fault test generation. They found that nearly 70% of these faults could have been discarded using timing window concept like ours.

Let us comment on the runtime of the proposed method to the runtimes of the tools using layout information in [10] and [11]. It is difficult to compare our experimental results to the experimental results in [10] and [11] fairly, because the tools in [10] and [11] were applied to only combinational circuits. Though neither provides the runtime information, we believe that the runtimes of our method are substantially shorter than those of the tools in [10] and [11]. We justify this based on the fact that the tools developed in both were applied only to small circuits. Also, it is pointed out in [10] that the expected test-generation runtime can be of the order of 40 s per fault. This implies that the runtime of their method is expected to be some order of magnitude higher than our method. Since we do not need detailed timing and layout information to identify the target faults, our method is much faster at the expense of some loss of accuracy. However, a careful choice of delay parameters will avoid false negatives in identifying false crosstalk faults in synchronous sequential circuits.

TABLE VI
ITC CIRCUIT EXPERIMENTAL RESULTS FOR IDENTIFYING TARGET FAULTS (Δ = 1 UNIT DELAY)

circuit names	No. of target faults (case 1)	No. of target faults (case 3)	No. of target faults (case 4)	target faults (%)	No. of false faults (case 1)	No. of false faults (case 2)	No. of false faults (case 3)	run times (sec.)	Improvement of the method (%)
b01	234	5	8	39	239	88	55	0.6	21
b02	236	8	6	49	154	54	52	0.4	52
b03	1,889	3	348	24	4,159	1,740	1,257	1.4	19
b04s	3,724	3	130	14	19,796	1,178	2,703	4.6	3
b06	323	4	8	49	207	54	86	0.7	18
b07s	2,274	3	144	14	12,522	1,236	1,761	3.3	5
b08	1,024	3	40	25	2,378	326	438	1.4	10
b09	5,465	6	216	43	4,583	1,264	1,786	2.0	44
b10	1,432	3	48	29	2,754	549	388	1.8	9
b11s	8,378	3	30	28	19,588	475	1,826	3.9	14
b12	10,200	6	960	18	37,134	8,240	5,681	8.8	3
b13s	1,845	3	156	26	3,731	987	898	3.0	3
b14s	33,281	3	244	4	835,199	4,721	45,077	34.9	2
b15s	72,333	3	1,344	4	1.833M	26,469	96,981	65.2	1
b17s	521,880	9	12,726	3	16.47M	216,873	997,566	200.9	1
b20s	63,319	3	489	4	1.609M	9,397	87,217	70.5	1
b21s	66,030	3	489	4	1.675M	9,781	87,217	73.6	1
b22s	102,238	3	734	4	2.606M	15,049	132,297	112.2	1

LP: Longest Path

M = 10^6

V. CONCLUSION

In this paper, we proposed a method for identifying the list of crosstalk-induced transition faults that may need to be tested in synchronous sequential circuits. We also identified the false crosstalk faults that need not (and/or cannot) be tested in synchronous sequential circuits. Our method, as presented in this paper, relies on the topological and timing information available at logic level to identify the pairs of aggressor and victim lines that may need to be tested. Experimental results for the gate level implementation of the ISCAS'89 and ITC'99 benchmark circuits show that the proposed method is very fast in obtaining the reduced lists of the target crosstalk faults and the lists are sufficiently smaller than the set of all possible combinations of faults.

If available, we believe that this method could also incorporate layout information, and the multiple aggressor requirements. Inclusion of such information should reduce the fault list even further. The layout information could be incorporated in Step 3 of Phase 1 of the proposed method (see Section III) by limiting only those lines to be in the target set, which are in physical or electrical proximity of each other. The multiple aggressor situation could be included in Phase 2 of the proposed method (see Section III) by comparing the timing window of a given victim node with the earliest and the latest transition times of multiple aggressor lines. However, evaluation of the quality of this method with layout information remains to be investigated and is beyond the scope of this paper.

REFERENCES

[1] N. Itazaki, Y. Idomoto, and K. Kinoshita, "A fault simulator method for crosstalk faults in synchronous sequential circuits," in *Proc. Int. Symp. Fault-Tolerant Comp.*, 1996, pp. 38–43.

[2] W. Y. Chen, S. K. Gupta, and M. A. Breuer, "Analytic method for crosstalk delay and pulse analysis under nonideal inputs," in *Proc. Int. Test Conf.*, 1997, pp. 809–818.

[3] ——, "Test generation in VLSI circuits for crosstalk noise," in *Proc. Int. Test Conf.*, 1998, pp. 641–650.

[4] ——, "Test generation for crosstalk-induced delay in integrated circuits," in *Proc. Int. Test Conf.*, 1999, pp. 191–200.

[5] ——, "Test generation for crosstalk-induced delay faults: Framework and computational results," in *Proc. Asian Test Symp.*, 2000, pp. 305–310.

[6] P. Agrawal, V. D. Agrawal, and S. C. Seth, "Generating tests for delay faults in nonscan circuits," *IEEE Des. Test Comput.*, vol. 10, no. 1, pp. 20–28, Jan. 1993.

[7] D. A. Kirkpatrick and A. L. Sangiovanni-Vincentelli, "Techniques for crosstalk avoidance in the physical design of high-performance digital systems," in *Proc. Design Automation Conf.*, 1994, pp. 616–619.

[8] P. Chen, D. A. Kirkpatrick, and K. Keuzer, "Switching window computation for static timing analysis in presence of crosstalk noise," in *Dig. Int. Conf. Computer-Aided Design*, 2000, pp. 331–337.

[9] K. J. Keller, H. Takahashi, K. K. Saluja, and Y. Takamatsu, "On reducing the target fault list of crosstalk-induced delay faults in synchronous sequential circuits," in *Proc. Int. Test Conf.*, 2001, pp. 568–577.

[10] S. Nazarian, H. Huang, S. Natarajan, S. K. Gupta, and M. A. Breuer, "Xiden: Crosstalk target identification framework," in *Proc. Int. Test Conf.*, 2002, pp. 365–374.

[11] A. D. Sathe, M. L. Bushnell, and V. D. Agrawal, "Analog macromodeling of capacitive coupling faults in digital circuit," in *Proc. Int. Test Conf.*, 2002, pp. 375–383.

[12] K. J. Keller, H. Takahashi, K. T. Le, K. K. Saluja, and Y. Takamatsu, "Reduction of target fault list for crosstalk-induced delay faults by using layout constraints," in *Proc. Asian Test Symp.*, 2002, pp. 242–247.

[13] L.-C. Chen, T. M. Mak, M. A. Breuer, and S. K. Gupta, "Crosstalk test generation on pseudo industrial circuits: A case study," in *Proc. Int. Test Conf.*, 2001, pp. 548–557.

[14] P. Chen, Y. Kukimoto, and K. Keutzer, "Refining switching window by time slots for crosstalk noise calculation," in *Dig. Int. Conf. Computer-Aided Design*, 2002, pp. 583–586.

[15] M. Abramovici, M. A. Breuer, and A. D. Friedman, *Digital Systems Testing and Testable Design*. Rockville, MD: Computer Science, 1990.

[16] P. Chen and K. Keutzer, "Toward true crosstalk noise analysis," in *Dig. Int. Conf. Computer-Aided Design*, 1999, pp. 132–135.

Hiroshi Takahashi (M'93) received the B.E. and M.E. degrees in electronic engineering from Saga University, Saga, Japan in 1988 and 1990, respectively, and the Dr. degree from Ehime University, Ehime, Japan, in 1996.

From 1997 to 1999, he was a Lecturer in the Department of Computer Science, Ehime University, where he has been an Associate Professor since 2000. From May 2000 to March 2001, he was a Research Fellow with the University of Wisconsin, Madison. His research interests are test generation and fault diagnosis for digital systems.

Prof. Takahashi has served as a Member of the IEEE Asian Test Symposium Program Committee.

Keith J. Keller received the B.S. degree in electrical and computer engineering from the University of Wisconsin, Madison, in 2000 and the M.S. degree in electrical engineering in 2002.

He is currently an Analog Power Supply Design Engineer with Texas Instruments, Dallas, TX.

Kim T. Le received the B.E. degree in electrical engineering from the National Institute of Technology, Saigon, Vietnam, in 1973, the M.E. in computer engineering from the Newcastle University, New South Wales, Australia, in 1986, and the Ph.D. from the Sydney University, Sydney, Australia, in 1991.

He has been a Lecturer at the University of Canberra, Canberra, Australia, since 1987. Previously, he was a Laboratory Supervisor with the National Institute of Technology, from 1973 to 1975, a Lecturer with the Polytechnique University, Ho Chi Minh City, Vietnam, from 1975 to 1981, a Commissioning Engineer with Eraring & Bayswater Power Stations, IHI Co., New South Wales, Australia, from 1982 to 1985, and a Tutor at Sydney University, from 1985 to 1987. His current research interests are in computer architecture design, implement and testing, image processing and analysis, and distributed systems.

Kewal K. Saluja (S'70–M'73–SM'89–F'95) received the B.E. degree in electrical engineering from the University of Roorkee, Roorkee, India, in 1967 and M.S. and Ph.D. degrees in electrical and computer engineering from the University of Iowa, Iowa City, in 1972 and 1973, respectively.

He is currently A Professor in the Department of Electrical and Computer Engineering at the University of Wisconsin, Madison, where he teaches courses in logic design, computer architecture, microprocessor-based systems, very large scale integration design and testing, and fault-tolerant computing. He was previously with the University of Newcastle, Newcastle, Australia. He has held visiting and consulting positions at various national and international institutions including the University of Southern California, Hiroshima University, the Nara Institute of Science and Technology, and the University of Roorkee. He has also served as a Consultant to the United Nations Development Program.

Prof. Saluja is a Member of Eta Kappa Nu, Tau Beta Pi, and a Fellow of the JSPS. He has served as an Editor of the IEEE TRANSACTIONS ON COMPUTERS from 1997–2001, and is an Associate Editor for the letters section of Kluwer's *Journal of Electronic Testing: Theory and Applications* (JETTA).

Yuzo Takamatsu (M'78–SM'99) received the B.E. degree in electrical engineering from Ehime University, Ehime, Japan, in 1966 and the Ph.D. degree from Osaka University, Osaka , Japan, in 1976.

He was a member of the Faculty of Science and Engineering, Saga University, Saga, Japan, from 1967 to 1987, where he was an Associate Professor in the Department of Electronic Engineering from 1975 to 1987. Since 1987, he has been a Professor in the Department of Computer Science, Ehime University. His research interests include test-pattern generation, fault simulation, and fault diagnosis.

Prof. Takamatsu is a Member of the Editorial Board of the *Journal of Electronic Testing: Theory and Applications* (JETTA). He served as the Program Chair of the 1994 IEEE Asian Test Symposium. He served as the Symposium Chair of the 1997 IEEE Asian Test Symposium. He served a Member of the IEEE Asian Test Symposium Program Committee.