Test Generation for Crosstalk-Induced Faults: Framework and Computational Results[1]

WEI-YU CHEN*
Sun Microsystems, Palo Alto, CA, 94303-4900, USA
weiyu.chen@sun.com

SANDEEP K. GUPTA and MELVIN A. BREUER
Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089-2562, USA
sandeep@poisson.usc.edu
mb@poisson.usc.edu

Received May 22, 2001; Revised Sep 15, 2001

Editor: Chauchin Su

Abstract. Due to technology scaling and increasing clock frequency, problems due to noise effects lead to an increase in design/debugging efforts and a decrease in circuit performance. This paper addresses the problem of efficiently and accurately generating two-vector tests for crosstalk induced effects, such as pulses, signal speedup and slowdown, in digital combinational circuits. These noise effects can propagate through a circuit and create a logic error in a latch or at a primary output. We have developed a mixed-signal test generator, called XGEN, that incorporates classical static values as well as dynamic signals such as transitions and pulses, and timing information such as signal arrival times, rise/fall times, and gate delay. In this paper we first discuss the general framework of the test generation algorithm followed by computational results. Comparison of results with SPICE simulations confirms the accuracy of this approach.

Keywords: Time-based test generation, fault modeling, mixed-signal test, crosstalk

1. Introduction

Due to the decrease in scaling of VLSI circuits, the increase in switching speeds, and the mixing of devices with different driving strengths, crosstalk effects are induced between some circuit elements [12, 13, 19, 20].

These effects can result in faulty behavior and hence become important issues in validation and testing. There are two main types of crosstalk effects: crosstalk induced pulses and crosstalk induced delay. The first creates a pulse on a line, called the victim line, which should remain in a static state when one or more neighboring lines, called affecting lines, have a transition. Depending on their amplitude and width, these pulses can have an important impact on circuit performance [17]. The second effect, crosstalk delay, is produced when both the affecting and victim lines have

[1] This work was supported in part by the Semiconductor Research Corp. under contract No. 98-TJ-646, and by Intel Corporation.

* Formally with Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089-2562

simultaneous or near simultaneous transitions. If both lines transit in the same direction, their transition times are reduced, hence the effective delay is reduced, leading to the phenomenon of *crosstalk speedup*. If the affecting and victim lines transit in the opposite direction, then there will be an increase in delay, referred to as *crosstalk slowdown*.

These changes in signal propagation delays can cause faulty behaviors. For example, many high performance circuits make extensive use of pipelines, shallow logic blocks between storage elements, dynamic gates, latches instead of flip-flops, single phase clocking, and performance based logic design and layout. The net result is that the timing margins between clocked elements are small. Hence delay must be well controlled and budgeted. Because crosstalk can adversely affect signal delay, coupling effects must be correctly handled to guarantee correct circuit operation. Thus the delay time for each combinational block and setup and hold times for latch elements must include the signal or clock skews due to crosstalk. Current trends in integrated circuit design indicate that signal noise and skew due to crosstalk can create severe design and test problems. Crosstalk effects coupled with “weak” spot defects create situations that must be addressed during testing. Also, process variations need to be considered in estimating the maximum value of these effects.

Although crosstalk coupling effects between interconnects have been previously studied [10, 17, 21, 23, 24], the focus has primarily been on crosstalk-induced pulses and related test generation techniques. Crosstalk–induced delay has received less attention. Logic level crosstalk fault models for pulses and PODEM based ATPG algorithms were presented in [16, 18, 20]. These models characterize crosstalk effects as static hazards having a full voltage swing, and result in an overestimation of noise strength. In addition, since crosstalk is a finite energy transient effect, test vectors generated using these models may not be able to actually propagate the noise to POs or flip-flops because of the inertia inherent to gates. The ability to *efficiently* and *accurately create* a *large* crosstalk effect and *propagate* it with *minimal attenuation*, and generate a test for crosstalk speedup/slowdown have not been previously addressed.

In this paper a mixed-signal test generation process is proposed where characteristics of crosstalk induced noise are accurately modeled. In addition to traditional logic values, the mixed-signal test generator also includes computation for analog properties such as noise strength (delay) and signal timings (such as arrival time and rise/fall times). New timing conditions are also proposed so that a large crosstalk effect, E, is first generated, and then other constraints employed so as to propagate the maximum effect of E, namely delay for slowdown or speedup, or amplitude and width for pulses, to an output or a flip-flop.

The paper is organized as follows. In Section II the theoretical foundations for the proposed methodology are presented. These results are used to compute analog properties of noise. In Section III various conditions and timing analysis for the creation of the worst-case coupling and propagation of the crosstalk signal are presented. These conditions are then incorporated into a test generation algorithm to generate tests for crosstalk–induced errors. Section IV shows experimental results of the proposed test generation algorithm. Finally, in Section V we present our conclusions.

2. Theoretical Foundation for the Proposed Test Generation methodology

Crosstalk is caused by parasitic couplings between adjacent wires that include inductive and capacitive effects. There is a low inductance value that becomes significant at very high frequency in certain lines, such as VDD and GND global buses, which are very long and wide (so R is comparable to ωL) and may conduct large switching current. For signal interconnects the capacitive coupling tends to dominate and it is still feasible to accurately model crosstalk without considering inductance because of the voltage-controlled nature of MOS devices, as detailed in [22]. Fig. 1 shows a simple circuit with coupling capacitance C_m between two signal lines A and V. The value of the parasitic capacitance C_m can be determined as described in [2, 5, 9].

To obtain insight into the detailed nature of crosstalk and its dependence on the circuit parameters associated with the coupled lines, we used the lumped model of capacitive coupling shown in Fig. 2. In this model each pulling resistance, R_{p1} and R_{p2}, is composed of the line resistance and the on-channel resistance associated with the line driver. The load capacitances, C_a and C_v, consist of the line capacitance and the gate capacitance of the load driven by the line. Thus the line driver is equivalent to a pulling resistance, and the coupling network can be viewed as a network of capacitors (C_m, C_a, C_v). This model allows for a somewhat general description of the signals A_{in} and V_{in}, not only in terms of their switching rates but also their relative skew.

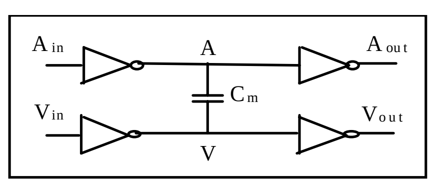

Fig. 1. Simple circuit showing source of crosstalk due to capacitive coupling.

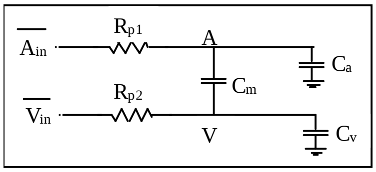

Fig. 2. Capacitive coupling model.

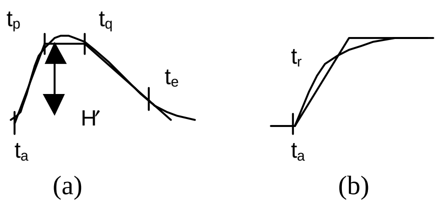

Fig. 3. Piece-wise-linear-model for noise: (a) pulse, (b) transition including speedup and slowdown.

By using Laplace transformations, we can obtain an expression for crosstalk in the s-domain, which we can transform back to the time domain [5]. The results obtained using this model are used in the development of our ATPG system. Many observations have been made based upon these analytic results such as the following: the faster the transition on the affecting lines, the larger the effect is on the victim line; the weaker the driver is on the victim line, the stronger the effect is on the victim line; the skew between the victim and affecting line transitions should be within one or two gate delays to have an appreciable impact. These and other conditions are incorporated into our test generation algorithm to maximize the crosstalk effect.

Once a noise is created at a fault site, we would like to characterize the noise and propagate it toward primary outputs to see if an error has been created. We use a piece-wise-linear model to characterize a noise waveform, as shown in Fig. 3. Here t_a is the arrival time of a signal and t_r is the rise time of a transition. Based on a circuit's electrical properties, we can estimate the height H′ of a crosstalk induced pulse as well as the parameters t_p, t_q and t_e that define its width.

Several techniques have been developed to propagate these piece-wise-linear input waveforms through CMOS gates to obtain output responses. These techniques include new models for a CMOS inverter, methods to calculate inverter output response for pulse inputs, and a method for collapsing CMOS gates into equivalent inverters. These techniques are then integrated into a test generation framework that takes into account several attributes such as noise strengths and signal arrival times and identifies test patterns that maximize crosstalk noise at POs while satisfying a given set of Boolean constraints. Full details of the piece-wise-linear noise model and the techniques mentioned above can be found in [6]. These models employ parameters charactering the gates in the circuits, such as transistor gain and load capacitances.

3. Test Generation for Crosstalk Faults

3.1. Test Generation Framework

In this section we present an ATPG algorithm to generate tests for crosstalk noise. This algorithm incorporates crosstalk by employing new logic values and corresponding analog information, such as signal arrival times, rise/fall times, and input arrival skews, and searches the space of all possible pairs of input patterns using a significant modified version of a backtrace procedure [11]. A signal value in our test generation system contains not only a symbol for its logic value, but also a set of parameters for its corresponding analog properties. Fig. 4 shows the framework of our ATPG system.

For a specific target crosstalk coupling in a circuit, the objective of this system is to generate, under given timing assumptions and requirements, a pair of vectors (a test) that create a crosstalk effect at the target and either a logic error or the maximum noise effect at an output. For example, in the case of crosstalk slowdown, given the timing of a clock-edge, the test generator may generate tests that cause a victim line signal to slowdown, and propagate the delayed signal in such a way so as to violate the given timing requirement at a D input to a flip-flop. For the proposed test generation framework new ATPG techniques were developed. A new value system that includes transitions and noise signals is used (see Table 1). The analytic models for the computation of the associated parameters are discussed in [6]. To create maximum crosstalk effects at primary outputs, conditions that help amplify the crosstalk noise were identified using the expressions developed in [5]. There are three objectives in creating a crosstalk effect of large severity: a weak driver on the victim line, a fast signal transition on the affecting line, and a propagation path that maintains/amplifies the noise effect until it reaches an output. These objectives are used to determine conditions to be satisfied for maximizing the observed crosstalk noise.

In Table 2 we list the conditions for each objective for a NAND gate. Similar conditions are established for other gate types. The objective line (affecting line, victim line, …) is assume to be driven by a NAND gate. Conditions in Table 2 are used by the backtrace process to select PI assignments that maximize crosstalk noise. For example, to propagate a delayed rising transition, denoted by S_dT_u through a NAND gate (line 9 in Table 2), we prefer to set all the side fan-in values to T_u, a rising transition. If not possible then some can be set to 1.

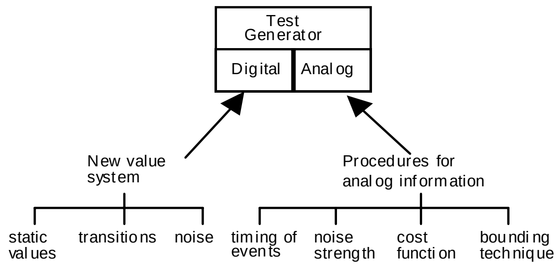

Fig. 4. The framework of our ATPG system.

Table 1. Symbols and parameters used for test generation.

Symbols	Associated parameters	Description
1	-	constant 1
0	-	constant 0
P_p	t_a, H′, t_p, t_q, t_e	positive pulse
P_n	t_a, H′, t_p, t_q, t_e	negative pulse
T_u	t_a, t_r	rising transition
T_d	t_a, t_f	falling transition
S_uT_u	t_a, t_r	Speedup rising transition
S_uT_d	t_a, t_f	speedup falling transition
S_dT_u	t_a, t_r	slowdown rising transition
S_dT_d	t_a, t_f	Slowdown falling transition
X	-	unknown

Description of parameters.
t_a - arrival time, H′, t_p, t_q, t_e as in Fig. 3 and defined in [6],
t_r - rise time, t_f - fall time

Table 2. Conditions for achieving three objectives (for a NAND gate).

Objective	Target value	Necessary conditions	Condition on other i/ps	
			Sufficient	Preferr-ed
Weak victim[1]	T_u	T_d at one i/p	1, T_d	all 1
Weak victim[1]	T_d	T_u at one i/p	T_u, 1	all T_u
Weak victim	0	1 at all i/ps	all 1	-
Weak victim	1	0 at one i/p	1, T_u, T_d, 0	1
Strong affecting[2]	T_u	T_d at one i/p	T_d, 1	all T_d
Strong affecting[2]	T_d	T_u at one i/p	1, T_u	all 1
Propagation	P_p/P_n	P_n/P_p at one i/p	1 when P_n/P_p arrives	all 1
Propagation[2]	S_uT_u	S_uT_u at one i/p	1, T_u	all 1
Propagation[2]	S_uT_d	S_uT_d at one i/p	T_d, 1	all T_d
Propagation[1]	S_dT_u	S_dT_u at one i/p	T_u, 1	all T_u
Propagation[1]	S_dT_d	S_dT_d at one i/p	1, T_d	all 1

1: all transitions preferred to be slow
2: all transitions preferred to be fast

Since the objective of this TG is to create the maximum noise at an output, in addition to the conditions in Table 2 we developed a cost function that can guide the search for PI assignments as well a path from the source of the noise to a PO. The cost function contains a digital and an analog part. The digital part deals with controllability and observability measures [1], and is used to break ties. The analog part of the cost function is a measurement of the gate's capability to propagate noise and is dependent on the gate's strength, i.e. effective β, and load capacitance. This cost function quantifies the "difficulty" which a pulse encounters in propagating through a gate. For instance, the load capacitance serves as a charge pool to mitigate the noise, and hence the larger the output capacitance the smaller the output pulse. On the other hand, the larger the β_{eff} the stronger the pull-down strength. Hence a small pulse can more easily discharge the output. For details of the derivation for cost functions please see 6.

After the analog cost of each gate is obtained, the cost of a path can be obtained by combining these cost values in a manner similar to calculation of observability costs. The computation of the analog cost of a path starts from the primary outputs, i.e. the last level of the gates, and the circuit is traversed backward to accumulate the cost of each gate. Thus, to propagate a noise effect we can first select a path whose cost is the lowest, i.e. propagates the noise with maximum severity. If two paths have the same analog costs, then the digital observability costs are used to break ties. In addition, another cost function exists that bounds the delay from a line to each primary output. This function is used to guide the propagation of a speedup or slowdown effect in such a way as to maximize the desired effect.

3.2. Timing-Oriented ATPG

A timing-oriented backtrace technique was developed [7, 8]. This technique enables the test generator to create a signal transition within a specific timing window, and helps to determine whether a crosstalk error can be created and/or detected. To excite a target effect at a specific time (or within a timing window), we first need to obtain some delay information about the gates and paths in the circuit. Related timing information is obtained through a process similar to static timing analysis and can be found in [3]. Recursive procedures are developed and used to guide the timing-oriented backtrace direction so that each objective in the test generation process can be satisfied under desired timing requirements. By using path delay information obtained from static timing circuit analysis, preferred paths can be selected during the backtrace and propagation processes. Fig. 5 shows an example of the timing-oriented backtrace process. Here due to a forward breadth-first processing of all gates, we can associate with each signal line α a timing window $[A_r^{min}, A_r^{max}]$, where A_r^{min} (A_r^{max}) is the minimum(maximum) arrival time of a rising transition on line α. Similarly we can associate information for a falling transition as well as upper and lower bounds on signal transition times. Assuming the inputs to a circuit change at time 0, let T be the minimum time after which a pulse or a slowdown effect at an output would create a problem. Then working back from the outputs of a circuit to its inputs, we can compute target timing windows. Such a window associated with a victim line indicates when a crosstalk effect must be generated to create an error at outputs after time T. In Fig. 5, let [z_1, z_2] be the target window at the output of gate g. The target window for line v_1 is as shown, where d_{1max} is the maximum delay of gate g, computed using the slowest input transition, and d_{1min} is the minimum delay computed using the fastest estimated transition. In [4], a more advanced timing model is described.

3.2.1. *Timing-Oriented Backtrace Procedure*

When an objective is processed, first we check for the existence of a compatible and incomplete pattern at the gate inputs. For example, consider an objective to have a falling transition at the output of gate g, as shown in Fig. 5. We check if the desired target timing window [z_1, z_2] overlaps with the timing windows [A_f^{min}, A_f^{max}] that we obtained from the timing analysis described in previous section. If not, then the desired objective cannot be achieved and a new objective must be selected. If the objective seems to be achievable, then for inputs having unknown value "X", we backtrace and search for a pattern to achieve the objective. For the input lines on which we select to backtrace, we compute the new target timing window [z_1-d_{1max}, z_2-d_{1min}]. Then the new target timing window is inserted into the new objective for the input we selected to backtrace, and we continue the backtrace process recursively.

The third step in processing an objective employs the condition that is used for side fan-in assignments. There are many patterns that can achieve a desired transition on a line with different transition times. For example, to create a falling transition at the output of a two-input NOR gate, both inputs having a rising transition will lead to a shorter gate delay than when one input has a rising transition and the other is held at constant 0. These conditions for side-fan-in assignments that help to create a faster transition were identified in 6. Table 2 (line 5 and 6) shows conditions for creating a fast transition at the output of a NAND gate.

3.2.2. *Incremental Timing Refinement*

Static timing analysis provides a min-max range for possible transitions on each line. The min-max range is due to unspecified input values. At each ATPG step, as more primary inputs get assigned values, more internal lines have known values and min-max timing ranges shrink due to recalculation of arrival, transition and required times. Hence as we dynamically update the timing information of signals, min/max timing ranges are refined to provide better timing information. Fig. 6 illustrates the idea of the output incremental timing refinement. Detail on incremental timing refinement can be found in [3].

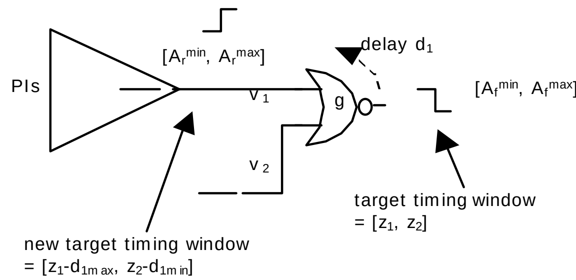

Fig. 5. Recursive execution of the backtrace process.

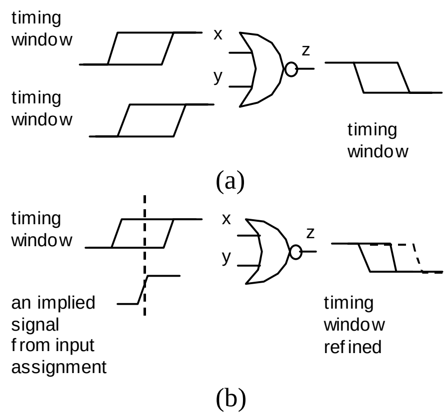

Fig. 6. Incremental timing refinement: a) before refinement, b) after refinement.

3.2.3. *Conflict between Objectives and Backtracking*

In an attempt to achieve the above objectives, there may be some occasions where some decisions made for earlier objectives block the chance of satisfying new objectives. Whenever these situations occur, backtracks are preformed until all objectives are achieved or it is determined that no test exists. For example, we may be able to achieve a fast affecting line transition by having all side fan-ins properly assigned, but this may make the desired transition on the victim line impossible to achieve. Hence an immediate backtracking is required to make an adjustment to the PI assignments for creating the affecting line transition so that the victim line transition can also be created. Backtracks may affect the quality of the resulting test, for example, they may lead to a stronger victim line driver. The algorithm employed will explore all possible PI combinations so that the best test, if one exists, will be eventually found. So, unlike PODEM which employs a constraint satisfaction search process, our algorithm attempts to maximize an objective, such as maximizing the delay (slowdown) of a signal transition. Because the proposed algorithm implicitly explores all PI combinations, it is necessary to limit the search space to improve efficiency. A bounding technique is proposed to reduce the search effort. It is only used for speedup and slowdown effects. So if a slowdown effect is propagated to a PO, its arrival time is recorded. The algorithm then continues in an effort to find a new effect that arrives even later than the current "latest" effect. The arrival time of a slowdown effect at a line g plus a lower bound on the gate delay along the shortest available path from g to a PO can be used to estimate whether or not to continue on the current path or to backtrack.

3.3. Test Generation Algorithm

The algorithm consists of five major steps to achieve the objectives. When a test is found, it is recorded and relevant signal information along the propagation path is stored to be used for branch-and-bound. Backtrack is performed to explore the search space until all PI combinations have been implicitly tried. The flowchart of the algorithm is shown in Fig. 7.

4. Experimental Results

4.1. Crosstalk Pulse

The test generation algorithm described was implemented in the C programming language and applied to several ISCAS '85 benchmark circuits. The program, called XGEN, was run on a Pentium II 400 MHz desktop. No circuit information, such as crosstalk fault locations, polarity of transitions causing crosstalk fault, coupling capacitance, and layout information, is currently available to us for these circuits. Thus the affecting and victim lines' driver strength and coupling capacitance value are assumed to be sufficient to excite a significant crosstalk noise at a fault site. We assume all transistors are 0.35um in length, the affecting line is driven by a large driver (28um PMOS/8um NMOS), the victim line is driven by a small driver (7um PMOS/2um NMOS), and they run parallel to each other for a distance of 1000um and have a coupling capacitance value of Cm = 300fF. All other gates and wires are assumed to have default device sizes and load capacitances.

Two sets of experiments were performed. In the first experiment a single crosstalk fault is targeted and XGEN is used to generate all possible tests for the target fault. Tests associated with corresponding pulses at POs are recorded so that the test creating the worst case pulse can be identified. The experimental results are shown in Table 3. In Table 3 PO denotes a primary output, first_p_amp is the height of the pulse at the fault site, and amp is the amplitude of the pulse at the corresponding output. Pulse amplitudes are normalized with respect to V_{DD}. The output statistics lumps the output pulses into voltage ranges. The results correlate well with SPICE simulations.

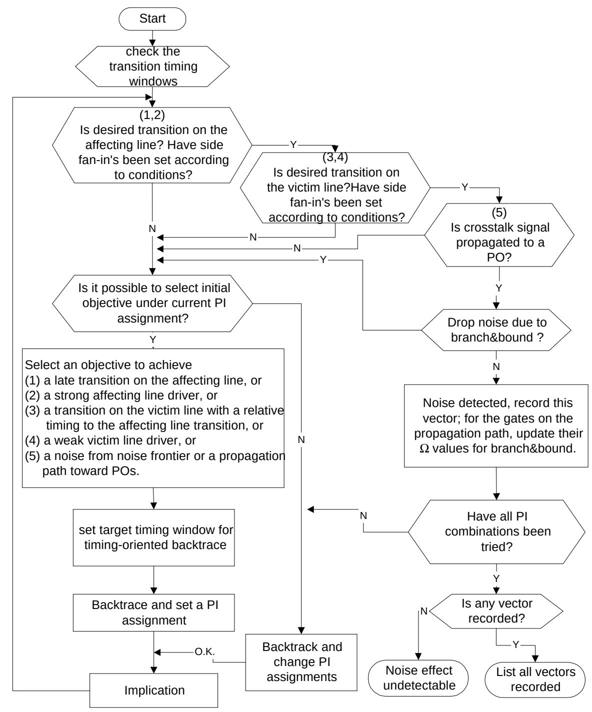

Fig. 7. Flowchart of the algorithm.

Table 3. Results of experiment 1: all tests for a single fault site.

Circuit c17.i
Affecting node 16 with rising transition,
Victim node 10 with value 0
Total 3 set of vectors: 12 out of 1024 combinations
$1T_d1T_uX$ first_p_amp=0.692 PO=22 type=P_n amp=0.897
$1T_d10X$ first_p_amp=0.642 PO=22 type=P_n amp=0.775
$111T_uX$ first_p_amp=0.659 PO=22 type=P_n amp=0.822
Output statistics
0.2-0.4Vdd 0.4-0.6Vdd 0.6-0.8Vdd >0.8Vdd
0 0 1 2
Total CPU run_time = 1 seconds

In the second experiment, for each circuit, 100 pairs of affecting and victim lines are selected at random without considering the circuit structure. A preprocessing step is performed so that the victim lines selected are located on critical paths. The proposed algorithm is applied to generate only one test for each fault. The maximum number of backtracks per fault is limited to 1000. The pulse size threshold at a PO is set to 0.2 V_{DD}, and any pulse smaller than the threshold will be ignored. Results of the experiments are shown in Table 4 and Table 5. In Table 4 there is no timing criterion set at primary outputs, and in Table 5 the longest path delay is set as the timing criterion at POs. For the latter case a pulse larger than 0.2 V_{DD} must occur at or after the specified time value for it to be considered a problem.

In Table 4 and Table 5, Column 2 shows the percentage of fault sites for which tests can be successfully generated. Column 3 shows the percentage of fault sites for which a test does not exist that propagates a crosstalk fault to a PO with significant amplitude (i.e. >0.2Vdd), and Column 4 shows the percentage of faults for which the number of backtracks exceeds the maximum setting and the TG process was aborted. Column 5 indicates the TG efficiency (Column 2 plus Column 3 divided by 100), and Column 6 is the total CPU time, expressed in seconds. As we can see from Table 5, if there is a timing criterion set at the primary outputs, then some large crosstalk effects that reach primary outputs may not actually violate the timing requirement. The process terminates when either 1) a crosstalk effect reaches a PO and violates the timing constraint, 2) the search space is exhausted hence no test exists, or 3) the backtrack limit is reached and the process is aborted. Therefore the percentage numbers in Column 3 and 4 increase, but the ATPG efficiency decreases. Since it takes time to search the PI space, the CPU time increases. Though not shown, as the value of the coupling capacitance is reduced, the percent detection also reduces. Unlike stuck-at test generation, one hopes that the percent of detected fault sites is zero.

Another experiment was performed to investigate the relationship between the detection rate and the threshold used to filter small pulses. The result is shown in Fig. 8. Fig. 8 shows that if we increase the threshold, some pulses that propagate to outputs are filtered away, and the percentage detection rate decreases. An obvious example is that if we set the threshold to be 1, then the detection rate becomes zero.

Although in the preceding experiments the device sizes, coupling capacitance, and related information are artificially inserted, the results in Table 4 and Table 5 demonstrate that the proposed algorithm can generate tests for circuits of reasonable sizes within an acceptable amount of time. That is, if all appropriate circuit and layout information is available, our algorithm can identify whether a significant crosstalk fault can be created and propagated to POs and generate an appropriate test. Since the execution of the proposed algorithm requires a non-trivial amount of calculation time, test pattern generation for all signal pairs of a complex circuit is not practical. Therefore, only critical pairs of lines should be targeted. The selection of these critical lines should be based on the circuit configuration, manufacturing process information, layout, designer's knowledge and other relevant information. This information is typically known in advance to the TG process, and should enable exclusion of many targets that cannot possibly cause errors at outputs. Thus, unlike stuck-at faults, we believe most circuits would have very few actual targeted crosstalk fault sites. In a separate piece of work, we are working on a preprocessor that prunes the space of targets to select a small set of potential targets that require process via XGEN.

Table 4. Result of experiment 2: one test for each fault site; Number of faults = 100; no timing criterion set at POs.

Circuit name	Successful TG (%)		TG Aborted (%)	ATPG Efficiency (%)	TG time (s)
	Detected	Undetect-able			
C432	33	56	11	89	1164
C880	41	46	13	87	1324
C1355	33	48	19	81	3866
C1908	50	34	16	84	2698
C2670	33	55	12	88	4542
C3540	29	49	22	78	4133
C5315	43	48	9	91	7090
C7552	31	58	11	89	7882
Ave.	36.625	49.25	14.125	85.875	4087

Table 5. Result of experiment 2: one test for each fault site; Number of faults = 100; the longest path delay is set as the timing criterion at POs.

Circuit name	Successful TG (%)		TG Aborted (%)	ATPG Efficiency (%)	TG time (s)
	Detected	Undetect-able			
C432	5	74	21	79	1246
C880	7	75	18	82	1648
C1355	5	70	25	75	3968
C1908	16	65	19	81	2508
C2670	7	72	21	79	4867
C3540	4	70	26	74	4614
C5315	11	74	15	85	7745
C7552	9	77	14	86	8651
Ave.	8.0	72.125	19.875	80.125	4406

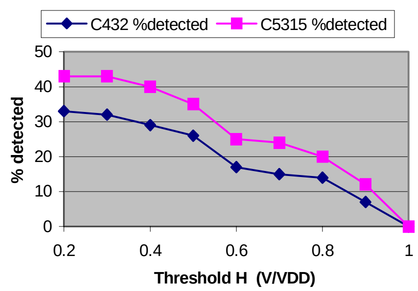

Fig. 8. Detection rate vs. pulse threshold.

4.2. Crosstalk Delay

Two sets of experiments were performed. In the first experiment a single crosstalk delay fault is targeted and the proposed algorithm used to generate all possible tests for the target. Tests associated with corresponding crosstalk delay that cause timing violations at POs are recorded so that the test creating the worst case timing violation at a PO can be identified. The results are shown in Table 6. All units are in pico second. The results correlate well with SPICE simulations. The timing criterion in Table 6 is the longest path delay of the circuit plus an extra delay of one gate delay, namely 335ps. If there is no crosstalk effect (C_m = 0), then there is no timing violation at any primary output. As we increase the coupling capacitance, the victim line signal becomes more delayed and its transition time increases. Also more test vectors can propagate the crosstalk delay signal to POs. This is because the delay slack is equivalently reduced and some vectors that could not cause timing violation before can do so now.

In the second experiment, for each circuit 100 pairs of affecting and victim lines are selected. If the selection of targets is completely random, then approximately 20 – 25 % of the targets have affecting and victim timing windows that do not overlap. Hence we preprocess the selection of targets so that the affecting and victim lines have overlapping timing windows. In addition, the victim lines are also located on critical paths so that a crosstalk effect propagating through these paths has a chance to cause a timing violation. XGEN is used to generate one test for each fault. The maximum number of backtracks per fault is again limited to 1000. Results of the experiments are shown in Table 7 (no timing criterion) and Table 8 (the longest path delay is set as the timing criterion). From Table 8 we again see that if there is a timing criterion set at the primary outputs then some crosstalk effects that reach primary outputs may not violate the timing requirement and hence become either undetectable crosstalk effects, or the TG aborts. The results in Table 7 and Table 8 again demonstrate that the proposed algorithm can generate tests for circuits of reasonable sizes, within an acceptable amount of time.

Another experiment was performed to see the impact of skew on the detection rate of crosstalk delay. The result is shown in Fig. 9. A skew of 1 implies that the transitions on the affecting and victim lines can be skewed for up to one gate delay, and a skew of zero means that both transitions have to switch simultaneously. Fig. 9 shows that as the skew increases, the detection rate increases because it increases the search space for test vectors. However, if transitions are far apart from each other, then there will be no crosstalk delay effect and hence the detection rate saturates.

A crosstalk delay signal can create a timing violation if there is not sufficient slack at the outputs. The following experiment was performed to study the amount of extra delay slack need to tolerate crosstalk delay. The result is shown in Fig. 10. The amount of increased delay at a fault site is from 30-120%, and the transition time increases from 10–110%. Because signal delay is accumulated along propagation paths, sufficient delay slack should be allocated at the outputs to avoid crosstalk slowdown causing a timing violation. Fig. 10 shows that for these example circuits with crosstalk effect at least two and half extra gate delays should be used to ensure correct circuit operations.

Table 6. Results of Experiment 1: all tests for a single fault. Circuit C17: affecting node 10 with a rising transition, victim node 16 with a falling transition; timing criterion = 335ps.

Tests	Crosstalk arrival time at the fault site (ps)	Crosstalk transition time at the fault site (ps)	Crosstalk arrival time at a PO (ps)
Cm = 0 fF; 0 test found			
Cm = 200 fF; 8 tests found			
T_d1T_d10	213	283	335
: (6 tests deleted) :			
$11T_d1T_d$	214	287	339
Cm = 300 fF; 16 tests found			
$T_dT_uT_dT_d0$	251	318	386
: (14 tests deleted) :			
$11T_d1T_d$	255	324	394

Table 7. Results of Experiment 2: one test for each fault site; number of faults = 100; no timing criterion set at POs.

Circuit name	Successful TG (%)		TG Aborted (%)	ATPG Efficiency (%)	TG time (s)
	Detected	Undetect-able			
C432	35	55	10	90	1019
C880	28	63	9	91	1553
C1355	16	67	17	83	3173
C1908	33	54	13	87	2562
C2670	17	74	9	91	4914
C3540	10	72	18	82	4565
C5315	31	59	10	90	7030
C7552	14	73	13	87	8424
Ave.	23.0	64.625	12.375	87.625	4155

Table 8. Results of Experiment 2: one test for each fault site; number of faults = 100; the longest path delay is set as the timing criterion at POs.

Circuit name	Successful TG (%)		TG Aborted (%)	ATPG Efficiency (%)	TG time (s)
	Detected	Undetect-able			
C432	15	68	17	83	1167
C880	13	72	15	85	1664
C1355	6	71	23	77	3403
C1908	15	70	15	85	2555
C2670	9	76	15	85	4870
C3540	4	72	24	76	4661
C5315	12	74	14	86	7323
C7552	7	75	18	82	8481
Ave.	10.125	72.25	17.625	82.375	4266

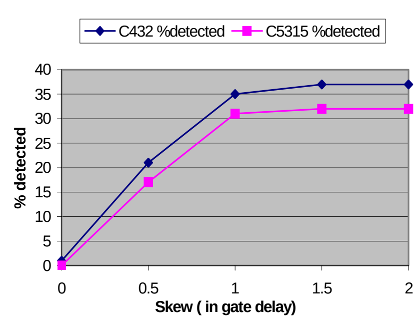

Fig. 9. Detection rate vs. skew between affecting and victim lines.

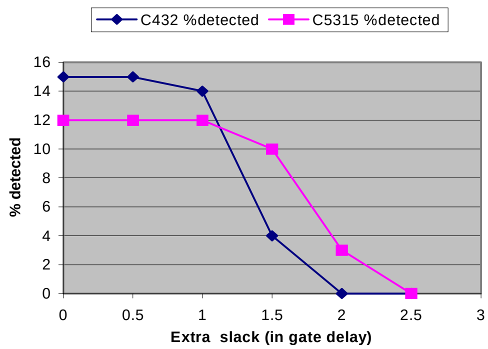

Fig. 10. Detection rate vs. extra delay slack.

5. Conclusions

Crosstalk effects can have a significant impact on signal integrity and delay. To ensure correct circuit operation, coupling effects, such as crosstalk-induced pulses, slowdown and speedup, should be taken into consideration in validating circuit designs and estimating the timing of critical paths.

We have presented a new test generation algorithm that not only considers speedup, slowdown and pulses as new logic values, but also takes into consideration information such as finite noise energy and input arrival skews to accurately characterize noise strength. The algorithm utilizes conditions that help excite the maximum crosstalk effect and propagate the crosstalk signal to POs under desired timing requirements. In addition, this algorithm includes the concept of gate delay, signal arrival time, signal strength and rise/fall times. By using the path delay information obtained by circuit preprocessing and/or the analog cost function, preferred paths can be selected during the backtrace and propagation processes. Because the proposed algorithm implicitly explores all PI combinations, it is beneficial to limit the search space to improve efficiency. A branch-and-bound technique is proposed to reduce the search effort. Finally, while most ATPG algorithms attempt to only satisfy a set of logical constraints, this algorithm also maximizes an objective function. Experimental results show that the method can be applied to selected crosstalk faults in circuits of reasonable sizes.

For crosstalk-induced delay, in our TG process a transition on the victim line is created and propagated along a possible long path, and receives the largest amount of crosstalk effect under certain timing requirements. From the point of view of both crosstalk effect and signal propagation, the amount of delay imposed on the victim line signal is maximized with respect to the given constraints. Hence the test vectors generated can be considered as a complementary set of tests for the purpose of delay testing. In the future we will address the issue of run time. Some techniques to be included in the next version of XGEN are multiple backtrace, dynamic cost re-computation, and backward implication with timing incorporated.

Reference

1. M. Abramovici, M. A. Breuer and A. D. Friedman, *Digital Systems Testing and Testable Designs*, IEEE Press, 1990.
2. N. D. Arora, K.V. Raol and R. Schumann "Modeling and extraction of interconnect capacitance for multilayer VLSI circuits", *IEEE Trans. on Computer-Aided Design and Integrated Circuits and Systems*, Vol. 15, No. 1, pp. 58-67, January 1996.
3. L. C. Chen, S. K. Gupta and M. A. Breuer, "A new framework for static timing analysis, incremental timing nalysis, and timing simulation," *Proc. Asian Test Symp.*, 2000, pp. 102-107.
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.*, June, 2001.
5. W. Y. Chen, S. K. Gupta and M. A. Breuer, "Analytic models for crosstalk delay and pulse analysis for non-ideal inputs," *Proc. Int'l Test Conf.*, 1997, pp. 809-818.
6. W. Y. Chen, S. K. Gupta and M. A. Breuer, "Test generation in VLSI circuits for crosstalk noise," *Proc. Int'l Test Conf.*, 1998, pp. 641–650.
7. W. Y. Chen, S. K. Gupta and M. A. Breuer, "Test generation for crosstalk-induced delay in integrated circuits," *Proc. Int'l Test Conf.*, 1999, pp. 191-200.
8. W. Y. Chen, M. A. Breuer and S. K. Gupta, "Timing analysis for test generation for crosstalk-induced delay in integrated circuits," Computer Engineering technical report No. 99-04, Electrical Engineering - Systems Department, University of Southern California, April 1999.
9. N. Delorme, M. Bellevile and J. Chilo, "Inductance and capacitance formulas for VLSI interconnects," *Electronic Letters*, Vol. 32, No. 11, pp. 996-997, May 1996.
10. D. S. Gao, A. T. Yang and S. M. Kang, "Modeling and simulation of interconnection delays and crosstalk in high-speed integrated circuits," *IEEE Trans. on Circuits and Systems*, Vol. 37, pp.1-9, January 1990.
11. P. Goel, "An implicit enumeration algorithm to generate tests for combinational logic circuits," *IEEE Trans. on Computer*, C-30, No.3, pp. 215- 222, 1981.
12. A. K. Goel and Y. R. Huang, "Modeling of crosstalk among the GaAs VLSI connections," *IEE Proc. Part G*, Vol. 136, pp. 361-368, 1989.
13. A. K. Goel, *High-speed VLSI Interconnections: Modeling, Analysis, and Simulation*, John Wiley & Sons Inc., 1994.
14. N. Hedebstierna and K. O. Jeppson, "CMOS circuit speed and buffer optimization," *IEEE Trans. on Computer Aided Design*, Vol. 6, pp.270-281, March 1987.
15. A. I. Kayssi, K. A. Sakallah, and T. M. Burks, "Analytical transient response of CMOS inverters," Trans. Briefs, *IEEE Trans. on Circuit and Systems*, Vol. 39, pp.43-45, January 1992.

16. K. T. Lee, C. Nordquist and J. A. Abraham "Automatic test pattern generation for crosstalk glitches in digital circuits," *Proc. VLSI Test Symposium*, 1998, pp. 34-39.
17. F. Moll and A. Rubio, "Spurious signals in digital CMOS VLSI circuits: a propagation analysis," *IEEE Tran. on Circuits and Systems –II: Analog and Digital Signal Processing*, Vol. 39, No. 10, pp. 749-752, October 1992.
18. F. Moll and A. Rubio, "Methodology of detection of spurious signals in VLSI circuits," *Proc. Europe Test Conference*, 1993, pp. 491-496.
19. A. Rubio and R. Anglada, "An approach to crosstalk effect analysis and avoidance techniques in digital CMOS VLSI circuits," *Int'l. Journal of Electronics*, Vol. 65, No. 1, pp. 3-17, 1988.
20. A. Rubio, N. Itazaki, X. Xu and K. Kinoshita, "An approach to the analysis and detection of crosstalk faults in digital VLSI circuits," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, Vol.13, No.3, pp.387-394, March 1994.
21. S. Voranantakul and J. L. Prince, "Crosstalk analysis for high-speed pulse propagation in lossy electrical interconnections," *IEEE Trans. on Components, Hybrids, and Manufacturing Technology*, Vol. 16, No. 1, pp. 127-136, February 1993.
22. R. K Watts, *Submicron Integrated Circuit*, NewYork: Wiley, pp. 317-318, 1989.
23. H. You and M. Soma, "Crosstalk and transient analysis of high-speed interconnects and packages", *IEEE Trans. on Solid State Circuits*, Vol. 26, pp. 319-30, March 1991.
24. A. E. Zain and S. Chowdhury, "An analytical method for finding the maximum crosstalk in lossless-coupled transmission lines," *Proc. Int'l Conf. on Computed Aided Design*, 1992, pp. 443-448.

Wei-Yu Chen received his Bachelor degree in Electrical Engineering from National Tsing-Hua University, Hsinchu, Taiwan, in 1992 and obtained MS degree in Electrical and Computer Engineering from University of California at Santa Barbara in 1994 and PhD degree in Electrical Engineering from University of Southern California in 2000. He is currently a member of technical staff at Processor and Product Group at Sun Microsystems, Sunnyvale, California. His research interests are in the area of noise modeling, test generation and validation for high performance circuits, at-speed testing, DFT, and physical design.

Sandeep Gupta received his Bachelors degree in Electrical Engineering from the Indian Institute of Technology, Kharagpur, India, in 1985 and obtained MS and PhD degrees in Electrical and Computer Engineering from the University of Massachusetts at Amherst in 1989 and 1991. He is an associate professor in the Department of Electrical Engineering-Systems at the University of Southern California, Los Angeles. He is also an associate editor of the *IEEE Transactions on Computers*. His research interests are in the area of VLSI testing and design. He is currently involved in projects on test and validation of deep submicron circuits, testing multicore systems-on-silicon, and delay testing and diagnosis of digital circuits. He is also involved in a project on testing and verification of network protocols. He is a recipient of the US National Science Foundation's Research Initiation Award (1992) and CAREER Award (1995). He is also a recipient of the Northrop Grumman Assistant Professorship (1995) and Zumberge Fellowship (1996) at the University of Southern California. He also received the Honorable Mention Award from the International Test Conference in 1997, and the co-author of the best paper at the 2000 Asian Test Symposium. He is a member of the IEEE Computer Society.

Melvin A. Breuer received his Ph.D. in Electrical Engineering from the University of California, Berkeley, is currently a Professor of both Electrical Engineering and Computer Science at the University of Southern California, Los Angeles, California, and is the Charles Lee Powell Professor of Electrical Engineering and Computer Science. He was Chairman of the Department of Electrical Engineering-Systems from 1991-1994, and is currently Chair once again. He was Chair of the Faculty of the School of Engineering, USC, for the 1997-98 academic year. His main interests are in the area of computer-aided design of digital systems, design-for-test and built-in self-test, and VLSI circuits.

Dr. Breuer is the editor and co-author of *Design Automation of Digital Systems: Theory and Techniques* (Prentice-Hall); editor of *Digital Systems Design Automation: Languages, Simulation and Data Base* (Computer Science Press); co-author of *Diagnosis and Reliable Design of Digital Systems* (Computer Science Press); co-editor of *Computer Hardware Description Languages and their Applications* (North-Holland); co-editor and contributor to *Knowledge Based Systems for Test and Diagnosis* (North-Holland); and co-author of *Digital System Testing and Testable Design* (Computer Science Press 1990 and reprinted in 1995 by the IEEE Press). He has published over 200 technical papers and was formerly the editor-in-chief of the *Journal of Design Automation and Fault Tolerant Computing*, on the editorial board of the *Journal of Electronic Testing*, the co-editor of the *Journal of Digital Systems*, and the Program Chairman of the Fifth International IFIP Conference on Computer Hardware Description Languages and Their Applications. He was co-author of a paper that received an honorable mention award at the 1997 International Test Conference, and the co-author of the best paper at the 2000 Asian Test Symposium. He is a Fellow of the IEEE; was a Fullbright-Hays scholar (1972); received the 1991 Associates Award for Creativity in Research and Scholarship from the University of Southern California, the 1991 USC School of Engineering Award for Exceptional Service, the IEEE Computer Society's 1993 Taylor L. Booth Education Award, and the first (2000) Engineering Faculty Council Award for Outstanding Meritorious Service to the USC School of Engineering. He was the keynote speaker at the Fourth Multimedia Technology and Applications Symposium, 1999, and the Ninth Asian Test Symposium, 2000.