Switch-level Delay Test

Suriyaprakash Natarajan Sandeep K. Gupta Melvin A. Breuer

EE-Systems, USC, Los Angeles, CA 90089-2562
{surya,sandeep,mb}@poisson.usc.edu

Abstract

Gate-level models are usually used to generate tests for circuits containing non-primitive CMOS gates. It is shown that tests generated using these models and classical conditions for robust path delay testing can fail to detect delay faults in such circuits. A new delay-independent, switch-level delay test methodology, called τ-robust testing, is proposed that defines new entities called targets *and proposes conditions to generate tests for each target. It is proven that, under the assumed delay model, a circuit that passes a test set containing a τ-robust test for every target is guaranteed to operate correctly at the desired speed. The effectiveness of the proposed methodology is demonstrated by (a) illustrating the difference between the delays excited by classical robust and τ-robust tests via circuit simulation, and (b) generation of τ-robust tests for benchmark circuits and comparison of τ-robust coverage of classical robust and τ-robust test sets.*

1 Introduction

The goal of delay testing is to verify the operation of a circuit at a specified clock rate. To ensure comprehensive delay testing, the path delay fault model has been developed [6, 11], which allows for the simultaneous presence of multiple delay faults distributed across various circuit lines. Furthermore, a comprehensive type of test, called robust test [6, 11], has been defined such that it guarantees detection of a delay fault on a target path independent of delay faults in other parts of the circuit.

The classical conditions for robust tests have been defined for circuits that contain only *primitive gates* (AND, OR, NAND, NOR, and NOT). However, most CMOS circuits contain many *complex CMOS gates*, such as various And-Or-Invert (AOI) gates and XOR gates. For purposes of delay test development, such a circuit is usually modeled using a circuit comprised of only primitive gates. This model circuit is then used, along with classical robustness conditions, for delay test generation and/or delay fault simulation.

It has been illustrated in [1] and [8] that there are differences between the behavior of a transistor-level circuit and that of its gate-level model. It has also been shown that the P and N networks of a complex CMOS gate can have structurally different primitive gate-level models and path delay tests developed using these gate-level models may not be reliable. Techniques for test generation for stuck-open and stuck-on faults using switch-level models have been reported in the literature ([3, 9, 10]). To date, we have been able to identify only one paper that deals with a switch-level algorithm for delay testing [1]. That work focuses on obtaining better accuracy in delay fault simulation using extensions of the classical robustness conditions to CMOS gates.

In the following, we demonstrate that for circuits containing complex CMOS gates, test sets obtained by using a circuit model comprised only of primitive gates and classical robustness conditions often fail to excite the worst case delays in the circuit. We show that this is also the case with tests generated using the robustness conditions proposed for CMOS gates in [1]. In fact, the delays excited by tests obtained using either of the above two approaches can be significantly lower than the critical circuit delays.

We generalize the concept of robust testing to make it applicable to circuits containing complex CMOS gates whose pull-up and pull-down networks contain transistors in arbitrary series-parallel configurations. We propose an enhanced theory and methodology for handling such circuits. The inadequacy of the basic entity that is currently tested in delay testing, viz., a *path*, is illustrated. We identify a new entity, called a *target*, to replace a path. We present the new *τ-robust* conditions that a test for a target must satisfy. We then define *functional sensitization* conditions for a target. We then prove that a test set that contains a τ-robust delay test for every functionally sensitizable target in a circuit guarantees, under the assumed delay model, complete testability of the circuit in the presence of delay faults.

It has been shown in [2] and [4] that the initial states of the capacitances internal to gates as well the numbers of inputs to on-path gates that have simultaneous transitions affect the delay excited for the target path. Currently, the proposed τ-robust methodology and the assumed delay model do not consider these phenomena. The extension of the τ-robust methodology to consider these delay phenomena is the subject of ongoing research.

The paper is organized as follows. Section 2 describes the delay model and the value system adopted in this paper. Section 3 provides the motivation for this work and describes the deficiencies of the existing methodology. Section 4 explains the problem that is addressed. An informal preview of the test generation methodology, followed by formal definitions and proof of completeness of methodology are provided in Section 5. Section 6 demonstrates the high quality of tests obtained with the new methodology, by circuit simulation on an example circuit, and by fault coverage results on complex gate versions of ISCAS89 benchmark circuits. The conclusions and scope of future work are given in Section 7.

2 Background

We now describe the delay model assumed and the value system used in this work.

2.1 Delay Model

The delay model assumes that a transistor is a perfect switch that switches instantaneously after its gate-to-source voltage crosses a specified threshold. The switching is also assumed to be independent of the switching of other transistors connected in series or parallel with the transistor. Furthermore, it is assumed that a transistor has a fixed resistance when conducting and infinite resistance when switched off. Capacitances internal to gates, crosstalk, and inductive effects are ignored. Finally, the short-circuit current in a CMOS gate is assumed to be zero.

2.2 Value System

A **pattern** $\mathcal{V}$ is defined as a sequence of two vectors (v_1, v_2) applied at the primary inputs of a circuit. We assume the slow-fast clocking methodology for the application of $\mathcal{V}$ where adequate time is allowed after the application of v_1 for the circuit to stabilize, even in the presence of any delay faults, before the application of v_2. The output is sampled after the system sampling period from the time of application of v_2.

We use a seven-valued algebra, **{T0,T1,S0,S1, fv0,fv1,XX}**, where $\mathbf{T}x$ ($x \in \{0,1\}$) implies a transition (with or without hazards) from $\overline{x}$ to x; $\mathbf{S}x$, a steady value of x with no hazards after the application of v_2; $\mathbf{fv}x$, a final value of x implied by v_2; and $\mathbf{X}$, a don't care. Also, a **change of state** is said to have occurred if the gate-to-source voltage of a transistor crosses its threshold causing it to switch from one state to another. The values applied at the primary inputs are assumed to be hazard-free.

3 Motivation

The traditional conditions for robustness of a test under the path delay fault model ensure only that at every gate along the target path, the output changes only after the first transition occurs at the on-path input of the gate. The conditions presented in [1] also ensure this for circuits comprised of complex CMOS gates. For a circuit comprised of primitive gates, any two-vector test that satisfies the above condition is deemed sufficient to test the path. This is because, under many of the commonly used delay models, all such tests excite the same delay at every gate along the target path. However, if tests are generated for a circuit with complex CMOS gates using an "equivalent" gate-level model, different robust tests that satisfy these conditions may excite different delays for the actual path being tested.

Consider the circuit shown in Figure 1(a) and one of its possible gate-level representations in Figure 1(b). Consider a path delay fault on the path PT shown in Figure 1(a) for a rising transition at its input n_2. Usually, the gate-level model (Figure 1(b)) and the classical conditions of robustness are used to obtain a test for the target fault PT. The conditions for robustly testing PT using the value system described in Section 2.2 are: a rising transition at n_2, $S0$ at n_3, and $fv1$ at n_7, as shown in the gate-level model in Figure 1(b).

Three classical robust tests, T1, T2, and T3, that satisfy the above conditions for the path PT are shown in Figure 1(c). Let the output be considered to have switched state when the output voltage crosses a specific threshold. Each test is shown on the gate-level equivalent circuit and also on the actual circuit. If we take the time constant for the charging/discharging of the output capacitance as a first-order delay metric, then tests T1 and T2 excite delays 4RC and 7RC, respectively, for propagating a rising transition through PT (as shown in Figure 1(c)). For test T3, depending on the arrival times of the transitions at the inputs of the corresponding N transistors, the delay of the transition could be anywhere between 3RC to 7RC. The delay excited will be 3RC if the output charge is conducted from its initial value continuously through the N transistor with input n_2 in series with the parallel connection of the N transistors with inputs n_4 and n_5. The delay will be 7RC if the output transition occurs by conduction through only the series connection of N transistors with inputs n_2 and n_5, i.e., the transition at n_4 arrives too late at the corresponding transistor input to be of any consequence with respect to switching of output node n_1.

The difference in delays excited is due to the fact that the delay is dependent on the discharge path within the complex CMOS gate. *This crucial information is lost in the gate-level equivalent circuit.* In general, it can be seen that if a circuit contains many complex gates

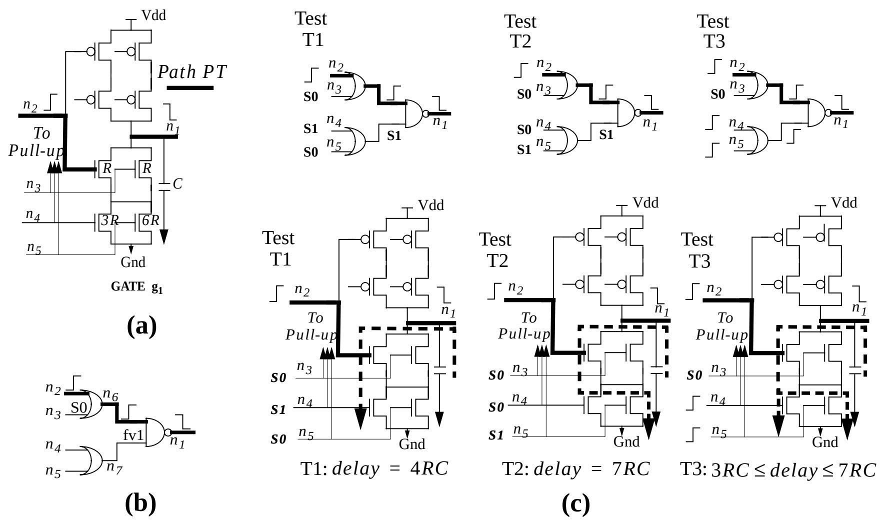


Figure 1. (a) Circuit with a complex CMOS gate, (b) gate-level model of (a), and (c) three classical robust tests for path PT in (a).

in sequence, then the difference in the delays excited through a path through these gates (1) by a robust test that switches on only a single series connection of transistors at every gate along the path, and (2) by another test that switches on multiple conduction paths at every gate along the path, will be very large, as the difference increases with each succeeding gate.

It is obvious from this example that, for circuits comprised of complex gates, all robust tests generated under the current methodology for a given fault do not excite equal delays. Since a typical delay test generator proceeds with the next fault whenever it finds *one* test for the current target fault, the resulting test thus obtained may not excite *all possible worst-case delays* of the target path. It should be noted that the switch-level delay test simulation technique proposed in [1] will also consider the above three tests as robustly sensitizing the path PT, since in all these tests, the N transistor in parallel to that driven by input n_2 is off, and transition at n_1 cannot occur unless the transition occurs at n_2.

This example illustrates that tests generated using a gate-level model of a circuit can fail to excite the worst-case delays in the actual circuit. This can occur for any target path for which the robustness conditions require a value at an off-path input in the gate-level model, such as node n_7 in Figure 1(b), where, (a) the value can be justified in multiple ways at the inputs of the gate driving the off-path input, and (b) the off-path input in the gate-level model and the gate driving that input are both parts of a single on-path complex CMOS gate in the actual circuit.

4 Problem Statement

Assuming an error-free design, any change in the temporal aspects of a circuit, due to manufacturing variations or defects, that causes an error to be sampled at the circuit outputs due to the application of a two-vector pattern is termed a delay fault in the circuit. For the delay model discussed above, a two-vector pattern is sufficient to detect a delay fault. Given a combinational circuit comprised of complex CMOS gates, the objective is to develop a methodology to generate tests that will guarantee detection of each delay fault in a circuit, independent of the delays in the rest of the circuit.

In the sequel, the circuit under test is assumed to have complex CMOS gates whose pull-up and pull-down structures contain transistors in arbitrary series-parallel configurations. Currently, the completeness of the proposed methodology has been proved under the delay model given in Section 2.1.

In the next section we will define the methodology and prove its completeness.

5 Proposed Test Methodology

5.1 Informal Preview

We now give a preview of the proposed methodology. The concepts will be formally defined in the subsequent sections.

In a circuit with primitive gates, a robust test for a logical path propagates the desired transition through the path while ensuring that at each gate along the path, the transition at the output of the gate occurs only after a transition at its on-path input. In a complex CMOS gate, a transition at an input can propagate in the manner mentioned above, to the output, in many ways depending upon the conducting paths that are established in the pull-up/pull-down network of the gate, as illustrated in Section 3. The output of a CMOS gate can be charged/discharged only through a single series connection of transistors or through a combination of multiple series connections in parallel. Hence, it may appear necessary to test for delays caused by conduction involving all possible combinations of connections of transistors in gates along a circuit path.

Even in the presence of delay faults, conduction through a single series connection of transistors in a gate is slower than conduction in parallel through multiple series connections including that single series connection.

Hence, for each gate in the circuit, we identify all such single series connections, called *strings*, in its pull-up and pull-down. An entity, called *target*, is constructed by selecting a *string* at each gate (from its pull-up or pull-down, whichever has to conduct for the propagation of the appropriate transition) along a circuit path, such that the string contains the transistor whose input is the on-path input of the corresponding gate. For a single circuit path, there can be many such *targets* depending on which string passing through the on-path input is selected at each gate along the path. We propose conditions to robustly test a target for a delay fault.

A CMOS gate is composed of an N and a P block. Since CMOS gates are inverting, the strings at consecutive gates along a target will be from opposite type blocks in the gates. The strings that lie on a target are called *on-target* strings and the inputs of the transistors that are on the path underlying the target are called *on-target nodes* of that target. Using the value system given in Section 2.2, we now propose the conditions to generate a test for a target for a rising (falling) transition at its input, and call the resulting test a τ-*robust* test (formal definition will follow in the subsequent sections). The test must provide:

1. a value of T1 (T0) at the input of the target,
2. a value of fv1 (fv0) at the inputs of each of the transistors that lie on every on-target string in an N block (P block), and
3. a value of S0 (S1) at the inputs of one or more transistors on each of the non on-target strings at every N block (P block) along the target.

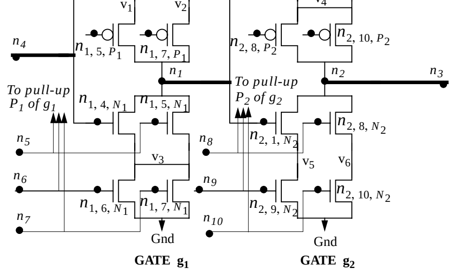


Figure 2. Circuit with complex CMOS gates.

Example 1: For the circuit shown in Figure 2, two targets associated with the path (n_4, n_1, n_2, n_3) for rising transitions at n_4 are shown in Figure 3. The conditions for generating τ-robust tests for those targets are also shown in Figure 3. There are four possible targets underlying the path mentioned above, for a rising transition at its input. □

In the subsequent sections, we formally describe this methodology. We also prove that under the assumed delay model, the proposed test conditions are sufficient to excite the worst-case delays of the circuit.

5.2 Definitions

A series-parallel complex CMOS gate, g_i, has two blocks, namely the pull-down, N_i, and pull-up, P_i, networks. Let $b_i \in \{N_i, P_i\}$. If a gate input fans out to multiple transistor inputs in the pull-down network (and hence also in the pull-up), then the fanout is considered external to the gate. Each primary input, primary output, gate input, transistor input (gate of transistor), and gate-output of a circuit is defined as a **node** (denoted by n). Every gate g_i has a unique output node n_i. Also, every input of a gate g_i is represented as $n_{i,j}$ such that $n_{i,j}$ is driven by the node n_j. The gate input node $n_{i,j}$ fans out to the N and P transistor for that input. The input node of a transistor of g_i that is on the branch of node $n_{i,j}$ in block b_i, $(b_i \in \{N_i, P_i\})$, is denoted by n_{i,j,b_i}. A primary input or a primary output is denoted by n_i (i is s.t. $\nexists$ gate g_i in the circuit). The transistor corresponding to the transistor input node n_{i,j,b_i} is denoted by $\mathbf{trans}(n_{i,j,b_i})$. The example shown in Figure 2 illustrates this notations (nodes of type $n_{i,j}$ are not shown in the figure to avoid clutter).

A **path, PT**, in a combinational circuit C is defined as a sequence that starts with a primary input, followed by a sequence of gate-output nodes, and ends with a primary output, $PT = (n_{i_0}, n_{i_1}, n_{i_2}, \ldots, n_{i_M}, n_{i_{M+1}})$, such that: n_{i_0} is a primary input connected to an input of gate g_{i_1}; n_{i_k} is the output node of gate g_{i_k} and is connected to an input of gate $g_{i_{k+1}}$; and $n_{i_{M+1}}$ is a primary output connected to n_{i_M}. The corresponding sequence $(g_{i_1}, g_{i_2}, \ldots, g_{i_M})$ is called the **gate trace, GT**, of the path PT. A circuit modeling technique (not given here) can be employed to ensure that every path and every primary output is uniquely specified by the above definition.

A **block trace, BT,** corresponding to a gate trace GT is defined as a sequence that starts with a primary input, followed by a sequence of blocks and a primary output, $BT = (n_{i_0}, b_{i_1}, b_{i_2}, \ldots, b_{i_M}, n_{i_{M+1}})$, such that: gates g_{i_k} lie on the gate trace GT of a circuit path starting at primary input n_{i_0} and ending at primary output $n_{i_{M+1}}$; n_{i_0} is a primary input connected to a transistor input node $n_{i_1,i_0,b_{i_1}}$ in block b_{i_1} of gate g_{i_1}; b_{i_k} and $b_{i_{k+1}}$ are opposite types of blocks of consecutive gates g_{i_k} and

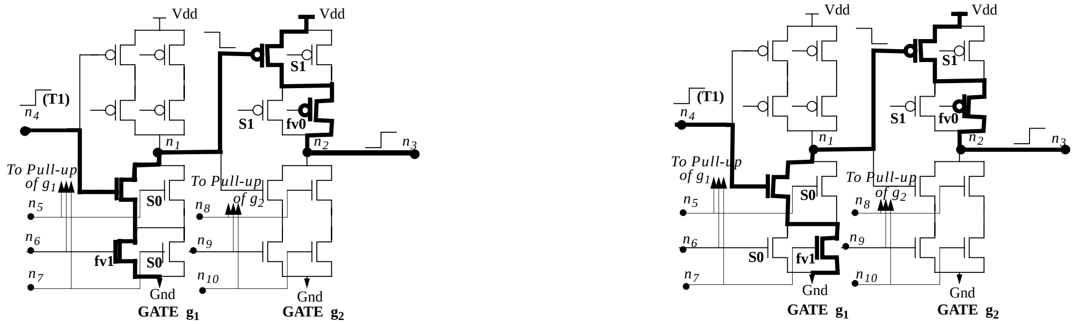


Figure 3. Two targets associated with the path in Figure 2 and their τ-robust test conditions.

$g_{i_{k+1}}$ in GT; and the output of gate g_{i_k} is connected to the transistor input node $n_{i_{k+1},i_k,b_{i_{k+1}}}$ in $b_{i_{k+1}}$. It can be seen that every circuit path has exactly two block traces depending on the choice of block for b_{i_1} in g_{i_1}. In a circuit comprised of primitive CMOS gates, a block trace specifies a logical path.

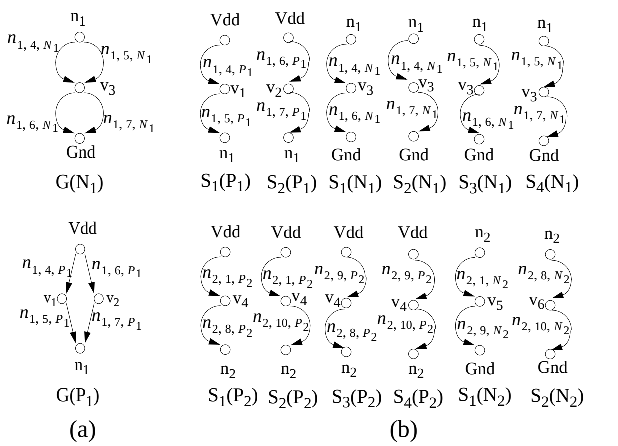
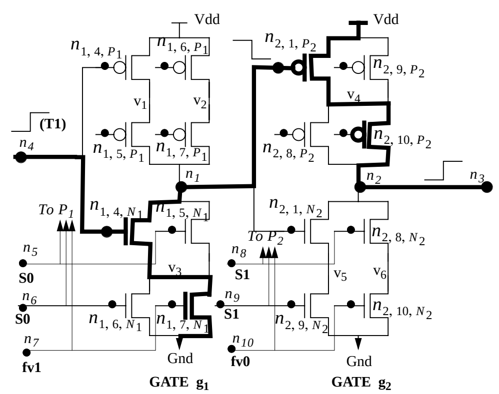


Figure 4. (a) Block graphs for gate g_1 in Figure 2, and (b) all strings in circuit in Figure 2.

A block b_i in a series-parallel complex CMOS gate g_i can be represented as a connected directed acyclic multigraph, called the **block graph** $G(b_i)[V, E]$. For $G(P_i)$ $(G(N_i))$, the internal terminals (drain/source terminals of transistors) in P_i (N_i), the output terminal of g_i, and the power terminal (ground terminal) form the vertices in the vertex set V; and the power terminal (output terminal) is termed the **origin vertex** and the output terminal (ground terminal) is termed the **destination vertex**. Every transistor in $G(P_i)$ $(G(N_i))$, $trans(n_{i,j,P_i})$ $(trans(n_{i,j,N_i}))$, that connects two vertices is modeled as a directed edge, with label n_{i,j,P_i}, and its direction is from source to drain (drain to source) of that transistor. All such edges form the edge set E. For the circuit in Figure 2, the graphs $G(P_1)$ and $G(N_1)$ for gate g_1 are shown in Figure 4(a).

A connected directed subgraph of $G(b_i)[V, E]$, formed by an alternating sequence of vertices and edges from the *origin* vertex to the *destination* vertex in $G(b_i)[V, E]$, such that no vertex or edge is repeated (called a *path* in graph theory terminology), is called a **string**, $S(b_i)[V', E']$. Given a transistor input node n_{i,j,b_i} in block b_i, let the set of all strings that pass through $trans(n_{i,j,b_i})$ be given by $\mathcal{S}(n_{i,j,b_i})$. All the strings in the gates in the circuit in Figure 2 are shown in Figure 4(b). A **target**, τ, is defined as a sequence comprised of a primary input, a sequence of strings, and a primary output, $\tau = (n_{i_0},\ S_{j_1}(b_{i_1}),\ S_{j_2}(b_{i_2}),\ \ldots,\ S_{j_M}(b_{i_M}),\ n_{i_{M+1}})$, such that: n_{i_0} is a primary input; $n_{i_{M+1}}$ is a primary output; blocks b_{i_k} are from a block trace, BT, starting at primary input n_{i_0} and ending at primary output $n_{i_{M+1}}$; and string $S_{j_k}(b_{i_k}) \in \mathcal{S}(n_{i_k,i_{k-1},b_{i_k}})$, where $n_{i_k,i_{k-1},b_{i_k}}$ is the transistor input node in b_{i_k} connected to the output of gate $g_{i_{k-1}}$ along BT. The transistor input nodes, $\{n_{i_1,i_0,b_{i_1}}, n_{i_2,i_1,b_{i_2}}, \ldots, n_{i_M,i_{M-1},b_{i_M}}\}$, are called the **on-target nodes** of τ. The strings on a target τ are called the **on-target strings** of τ.



Target τ_5 = (n_4, $S_2(N_1)$, $S_2(P_2)$, n_3)

Input order: n_4, n_5, n_6, n_7, n_8, n_9, n_{10}

A τ-robust test for τ_5 = (v_1, v_2) = (T1, S0, S0, fv1, S1, S1, fv0)

Figure 5. Target τ_5 for circuit in Figure 2 and a τ-robust test for τ_5

τ_1 = (n_4, S_1(P1), S_1(N2), n_3)
τ_2 = (n_4, $S_1(N_1)$, $S_1(P_2)$, n_3)
τ_3 = (n_4, $S_1(N_1)$, $S_2(P_2)$, n_3)
τ_4 = (n_4, $S_2(N_1)$, $S_1(P_2)$, n_3)
τ_5 = (n_4, $S_2(N_1)$, $S_2(P_2)$, n_3)
τ_6 = (n_5, $S_1(P_1)$, $S_1(N_2)$, n_3)
τ_7 = (n_5, $S_4(N_1)$, $S_1(P_2)$, n_3)
τ_8 = (n_5, $S_4(N_1)$, $S_2(P_2)$, n_3)
τ_9 = (n_5, $S_3(N_1)$, $S_1(P_2)$, n_3)
τ_{10} = (n_5, $S_3(N_1)$, $S_2(P_2)$, n_3)
τ_{11} = (n_6, $S_2(P_1)$, $S_1(N_2)$, n_3)
τ_{12} = (n_6, $S_1(N_1)$, $S_1(P_2)$, n_3)
τ_{13} = (n_6, $S_1(N_1)$, $S_2(P_2)$, n_3)
τ_{14} = (n_6, $S_3(N_1)$, $S_1(P_2)$, n_3)
τ_{15} = (n_6, $S_3(N_1)$, $S_2(P_2)$, n_3)
τ_{16} = (n_7, $S_2(P_1)$, $S_1(N_2)$, n_3)
τ_{17} = (n_7, $S_2(N_1)$, $S_1(P_2)$, n_3)
τ_{18} = (n_7, $S_2(N_1)$, $S_2(P_2)$, n_3)
τ_{19} = (n_7, $S_4(N_1)$, $S_1(P_2)$, n_3)
τ_{20} = (n_7, $S_4(N_1)$, $S_2(P_2)$, n_3)
τ_{21} = (n_8, $S_1(P_2)$, n_3)
τ_{22} = (n_8, $S_3(P_2)$, n_3)
τ_{23} = (n_8, $S_2(N_2)$, n_3)
τ_{24} = (n_9, $S_3(P_2)$, n_3)
τ_{25} = (n_9, $S_4(P_2)$, n_3)
τ_{26} = (n_9, $S_1(N_2)$, n_3)
τ_{27} = (n_{10}, $S_2(P_2)$, n_3)
τ_{28} = (n_{10}, $S_4(P_2)$, n_3)
τ_{29} = (n_{10}, $S_2(N_2)$, n_3)

Figure 6. Target set for circuit in Figure 2.

Example 2: All the targets in the circuit in Figure 2 are listed in Figure 6. The target τ_5 is shown in Figure 5. □

5.3 Test Methodology

We propose testing *targets* to detect delay faults in a circuit. The delay model is as given in Section 2.1.

5.3.1 Test Generation for Targets

In this section, we define the conditions that a delay test for a target must satisfy and illustrate the test generation, while deferring to the subsequent sections the proof of desirable properties of such tests.

A pattern $\mathcal{V} = (v_1, v_2)$ applied to a circuit is said to have **τ-propagated** a specified transition through a target $\tau = (n_{i_0}, S_{j_1}(b_{i_1}), S_{j_2}(b_{i_2}), \ldots, S_{j_M}(b_{i_M}), n_{i_{M+1}})$, if

1. it provides a rising (falling) transition at the primary input n_{i_0}, if $b_{i_1} = N_{i_1}$ (P_{i_1}),
2. the first change of state at $trans(n_{i_k,i_{k-1},b_{i_k}})$ occurs only after the first change of state at $trans(n_{i_{k-1},i_{k-2},b_{i_{k-1}}})$, for $k = 1, 2, ..., M+1$, where $n_{i_0,i_{-1},b_{i_0}} = n_{i_0}$ and $n_{i_{M+1},i_M,b_{i_{M+1}}} = n_{i_{M+1}}$, and
3. the first change of state at the on-target node, $n_{i_{k+1},i_k,b_{i_{k+1}}}$, occurs through exclusive conduction via the on-target string $S_{j_k}(b_{i_k})$ in b_{i_k}.

A pattern $\mathcal{V}$ that *τ-propagates* the relevant transition through a target τ is a delay test for τ.

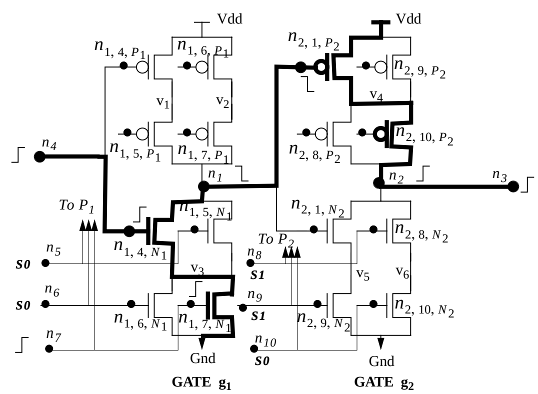
We first enumerate all the targets for the given circuit. For each target τ, we apply the *t-propagation* conditions at every gate along the gate trace associated with τ to obtain a delay test $\mathcal{V}$ for τ. For brevity, we discuss only the conditions for an N block. The conditions for a P block are similar. To obtain a delay test for a target τ, at every gate g_{i_k} along the gate trace associated with τ, if the on-target string is $S_{j_k}(N_{i_k})$, then,

1. a value of $T1$ must be justified at the on-target node, $n_{i_k,i_{k-1},N_{i_k}}$, of τ in g_{i_k},
2. a value of $fv1$ must be justified at each of the input nodes $n_{i_k,l,N_{i_k}}$ of g_{i_k}, where $n_{i_k,l,N_{i_k}}$ is such that $trans(n_{i_k,l,N_{i_k}})$ is on the string $S_{j_k}(N_{i_k})$, and,
3. a value of $S0$ must be justified at the input nodes $n_{i_k,l',N_{i_k}}$ of one or more edges, $trans(n_{i_k,l',N_{i_k}})$, on each of the other strings in $G(N_{i_k})$.

Example 3: For the target τ_5 shown in Figure 2, the conditions for test generation and a test $\mathcal{V}$ that satisfies these conditions are shown in Figure 5. □

5.3.2 Target Delay and Faulty Target

In this section, we will define the delay of a target and the concept of a faulty target. Certain definitions are required for this purpose. The circuit model is as given in Section 2.1. For simplicity, we discuss only the falling transition at the output of a gate (and hence analyze only its N block) and provide definitions only relevant to these. The definitions for a rising transition at the output of a gate are similar.



Target τ_5 = (n_4, $S_2(N_1)$, $S_2(P_2)$, n_3)

Figure 7. Illustration of string delay.

In a circuit C, after a vector v has been applied and C allowed to stabilize, the primary input and the output node of every gate holds a certain amount of charge corresponding to a particular logic value that appears on that node. This charge is the sum of the charge on each capacitance connected to the node after C has stabilized for v. Given a particular logic value L, $L \in \{0, 1\}$, and circuit realization, this charge has a fixed value for a given node n_i in C. Let this value be called the **extremum charge**, $Q(L, n_i)$. Now, consider a string $S = S_{j_k}(b_{i_k})$ in a target τ. Let $b_{i_k} = N_{i_k}$. Let the node n_{i_k} hold the extremum charge $Q(1, n_{i_k})$. The time required for a change of state to occur at the succeeding on-target node in τ, $n_{i_{k+1},i_k,P_{i_{k+1}}}$, by continuous and exclusive conduction through the string S in gate g_{i_k} after initiation of conduction is called the **string delay**, $SD(\tau, S)$, of the string S on τ. Note that for a given fabricated instance of the circuit C, $SD(\tau, S)$ is a constant and is a consequence of the physical properties of C.

Example 4: Consider the gate g_1 in Figure 7. Let a pattern $\mathcal{V}$ (shown in the figure) imply values T1 (without hazard) at inputs n_4 and n_7, and $S0$ at inputs n_5 and n_6. This would cause a change of state at $trans(n_{2,1,P_2})$. Assume that, on target τ_5 shown in Figure 7, the change of state at $trans(n_{1,7,N_1})$, due to the transition at $n_{1,7,N_1}$, occurs later than that at $trans(n_{1,4,N_1})$ The string delay, $SD(\tau_5, S_2(N_1))$, is given by the time delay between the change of state at $trans(n_{1,7,N_1})$ and the change of state at $trans(n_{2,1,P_2})$. □

The **target delay,** d_τ, of a target $\tau = (n_{i_0}, S_{j_1}(b_{i_1}), S_{j_2}(b_{i_2}), \ldots, S_{j_M}(b_{i_M}), n_{i_{M+1}})$ is defined as the sum

$$d_\tau = D(n_{i_0}, n_{i_1, i_0, b_{i_1}}) + \Sigma_{k=1}^{M} SD(\tau, S_{j_k}(b_{i_k})),$$

where $SD(\tau, S_{j_M}(b_{i_M}))$ is defined with respect to a change of state at primary output $n_{i_{M+1}}$, and $D(n_{i_0}, n_{i_1, i_0, b_{i_1}})$ is the time difference between a change of state from 1 to 0 (0 to 1) at primary input n_{i_0} to the corresponding change of state at $n_{i_1, i_0, b_{i_1}}$, if $b_{i_1} = P_{i_1}$ ($b_{i_1} = N_{i_1}$).

A target τ is said to be **faulty** due to the presence of delay faults if the *target delay*, d_τ, of τ, exceeds the system sampling period.

5.3.3 τ-robust delay test

We now formally define a **τ-robust** delay test for a target in terms of the desirable robustness properties it must satisfy.

A pattern $\mathcal{V} = (v_1, v_2)$, is said to be a **τ-robust** delay test for a target τ, if, when τ is faulty due to the presence of delay faults and pattern $\mathcal{V}$ is applied, the value at the circuit output is guaranteed to be different from the expected value at the sampling time, independent of the delays in other parts of the circuit.

We now relate the τ-robustness property that we desire in a delay test to the τ-propagation conditions we proposed in Section 5.3.1, by the following theorem.

Theorem 1 *If a pattern $\mathcal{V} = (v_1, v_2)$* **τ-propagates** *the relevant transition through a target τ, then $\mathcal{V}$ is a* **τ-robust** *delay test for τ.*

Proof: Proof of this theorem can be found in [7]. □

Thus, the test shown in Example 3 in Section 5.3.1 for target τ_5 in Figure 5, is a τ-robust test for τ_5.

5.3.4 Functional Sensitization of Targets

A target $\tau = (n_{i_0}, S_{j_1}(b_{i_1}), S_{j_2}(b_{i_2}), \ldots, S_{j_M}(b_{i_M}), n_{i_{M+1}})$, is said to be **functionally sensitizable** if there exists a vector v that provides a value of 1 (0) at the input node $n_{i_k, l, N_{i_k}}$ ($n_{i_k, l, P_{i_k}}$) of every transistor, $trans(n_{i_k, l, N_{i_k}})$ ($trans(n_{i_k, l, P_{i_k}})$), on the on-target string $S_{j_k}(N_{i_k})$ ($S_{j_k}(P_{i_k})$) of τ. In other words, τ is said to be functionally sensitizable if there exists a vector such that it provides a value at the input of each of the transistors on every on-target string of τ such that the transistors conduct. Targets that are functionally unsensitizable do not affect circuit delay.

5.3.5 Completeness of Methodology

A circuit C is said to be **completely τ-robustly testable** if there exists a τ-robust test for every functionally sensitizable target in C. A test set T is said to be a **complete τ-robust test set** for a circuit C if T contains a τ-robust test for every functionally sensitizable target in C. (Note that a *complete τ-robust test set* exists only for a *completely τ-robustly testable* circuit.) A circuit C is said to be **delay-fault-free** if there does not exist a two-vector pattern $\mathcal{V}$ which, when applied to C, causes C to fail due to delay faults. The following theorem establishes the completeness of our methodology.

Theorem 2 *A completely τ-robustly testable circuit C is* **delay-fault-free** *if C* **successfully passes** *a* **complete τ-robust test set** *T for C.*

Proof: Proof of this theorem can be found in [7]. □

Note that a τ-robust test for a target is also a robust test for the associated logical path. All targets of logical paths which are robustly untestable are τ-robustly untestable. And all targets of logical paths that are identified as functionally unsensitizable (according to functional sensitization conditions at the path level of abstraction) are functionally unsensitizable according to our definition. However, there may exist targets of functionally sensitizable logical paths, that are functionally unsensitizable.

For circuits with only primitive gates, τ-robust theory reduces to the classical robust theory. In that case, a target reduces to its underlying logical path, a τ-robustly testable target reduces to its underlying robustly testable logical path, and a functionally sensitizable target reduces to its underlying functionally sensitizable logical path.

6 Experimental Results

6.1 Delays Excited by τ-robust Tests

We first illustrate the effectiveness of the τ-robust delay test methodology with the circuit shown in Figure 2. This circuit has 29 targets (Figure 6). Its gate-level model is shown in Figure 8, which, according to the classical robust delay test terminology, contains 14 logical paths. A complete classical robust test set and

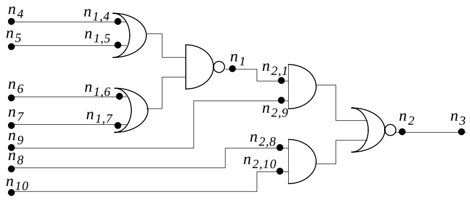


Figure 8. Gate-level model for circuit in Figure 2.

a complete τ-robust test set were generated using the gate-level model and the switch-level model, respectively. The delay between the time at 50% value of the input transition(s) and the time at 50% value of the output transition in a delay-fault-free version of the circuit due to each of these tests is determined by simulation using HSPICE employing 0.8μ CMOS process parameters. Each transistor used in the circuit has $W = 3.2\mu$ and $L = 0.8\mu$. The delay distributions of the two test sets are shown in Figure 9.

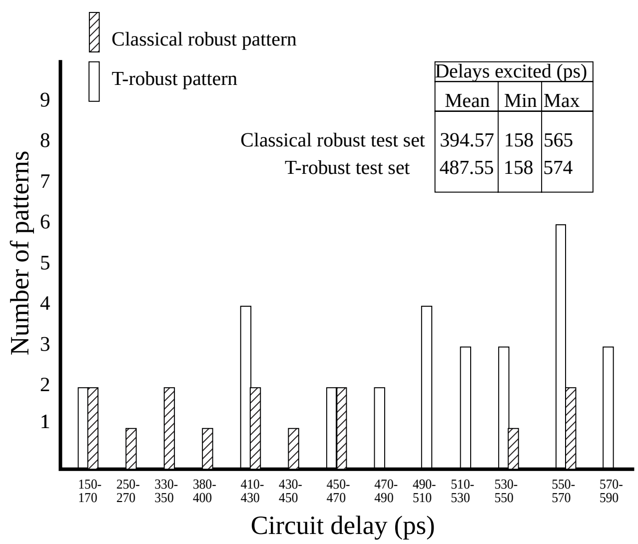


Figure 9. Delay distributions of classical robust and τ-robust test sets.

It is observed that the τ-robust test set excites a higher maximum delay and also a significantly higher average delay compared to the classical robust test set. Furthermore, the delays excited by the τ-robust test set are clustered near the maximum delay value and hence expose the critical path delays whereas this is not the case with the classical robust test set. Since the phenomena that causes the difference in delays excited by the classical and τ-robust test sets is a first-order effect, the difference in test quality is significant. Thus, the tests from the classical test set can be ineffective in exercising a significant number of critical paths resulting in chips with delay faults being sent to the customer.

6.2 τ-robust Test Generation for Benchmark Circuits

A target is said to be **testable** if there exists a τ-robust test for the target. The set of all targets in a circuit constitutes the universe of faults. The **τ-robust fault coverage** of a test set is defined as the fraction of the total number of targets that are τ-robustly tested by the test set. The **testable τ-robust fault coverage** of a test set is defined as the fraction (expressed as a percentage) of the number of targets detected by the test set to the total number of testable targets in the circuit.

Experiments were conducted as follows. For each ISCAS89 circuit C, a new circuit, C^*, comprising of complex CMOS gates with series-parallel transistors in their pull-up/pull-down networks was created. The boolean functionality of C^* is the same as that of C, and if each of the complex CMOS gates in C^* is replaced by an equivalent canonical representation using primitive CMOS gates, then the resulting structure is identical to that of C. In other words, most current tools that perform gate-level modeling of complex CMOS gate circuits for purposes of delay test generation would convert C^* to C.

τ-robust test generation for targets can be performed on either a switch-level netlist or an augmented gate-level netlist that contains information about the structure of CMOS gates. We adopted the second approach. Due to space constraints, this approach is not detailed here.

For each benchmark circuit C (containing only primitive gates), a classical robust test set T' and its robust fault coverage were obtained. τ-robusts tests were then generated for targets in the corresponding complex gate version C^* to obtain a test set T. τ-robust fault simulation with the patterns in T' was then performed on C^* to obtain its τ-robust fault coverage. T and T' were then compared based on their τ-robust fault coverages.

Table 1 shows the results obtained for some ISCAS89 benchmark circuits. The legend for reading the columns is as follows: CG - number of complex gates; P - total number of paths in benchmark circuit C; TP - number of robustly testable paths in C; τ - total number of targets in circuit C^*; τ_LP - number of targets whose τ-robust conditions in C^* are identical to the classical robust conditions for the corresponding logical paths in C; τ_FUS - number of functionally unsensitizable targets in C^*; $T\tau(T)$ - number of targets τ-robustly testable by T; $T\tau\_LP(T)$ - number of targets testable by T whose τ-robust conditions are identical to the classical robust conditions for the corresponding logical paths; $T\tau\_NLP(T)$ - number of targets testable by T whose τ-robust conditions are not identical to the classical robust conditions for the corresponding logical paths, $UT\tau(T)$

- number of targets τ-robustly untestable by T. The notations are similar for test set T'.

It should be noted that due to the practical difficulties in obtaining the C^* circuits with many complex CMOS gates, and due partly to the nature of the benchmark circuits themselves, only a few complex gates are present in each C^* circuit as shown in Column 2 of Table 1. Also, due to the above reasons, the complex gates obtained do not possess rich series-parallel structures. Hence, a major portion of the structure of each C^* circuit is identical to that of its corresponding benchmark circuit C, as seen by the significant numbers for τ_LP (Column 6) in Table 1.

Notice that for the benchmarks circuits shown, the percentage of τ-robustly testable targets is less than the percentage of robustly testable paths in the primitive gate level equivalent circuit. Such a comparison is, however, not advisable since the fault coverages are for different fault models (targets or paths). It would be appropriate to compare the testable τ-robust fault coverage of the classical robust test set with that of the generated τ-robust test set to evaluate their effectiveness with respect to testing the targets. The drop in testable fault coverage with the classical robust test set (given by $(100 \times ((T\tau(T) - T\tau(T'))/T\tau(T))$ and given in Column 14 of Table 1) is on an average 10.6%. The maximum drop in testable fault coverage is about 20% for s420 and the minimum drop is about 1.75% for s820. The drop in fault coverage would be more significant for circuits that contain numerous complex CMOS gates with rich series-parallel structures, than for the versions of ISCAS89 circuits considered.

To save space, $T\tau\_LP(T')$ is not shown in Table 1, but our results indicate that $T\tau\_LP(T') = T\tau\_LP(T)$. This should be no surprise since the τ-robust theory reduces to the classical robust theory in this case. Also, for most circuits, $T\tau\_LP(T)$ is significant. This is because τ_LP is significant (the reason for which was explained earlier), and the original C circuits are highly robustly testable. However, if we consider only those targets in C^* for which the τ-robust test generation conditions are different from the conditions for classical robust test generation of their corresponding logical paths in C, the drop in testable coverage for a classical robust test set, given in Column 15 of Table 1, is on the average 66%. The maximum drop in this coverage is 100% and occurs for circuits s208 and s420 and the minimum drop is 34.6% which occurs for circuit s641. Also, note that the number of functionally unsensitizable targets is also significant for some circuits (Column 7, τ_FUS, in Table 1).

It should be noted that any target that is not τ-robustly tested by a classical delay test set may be a critical path of the circuit with a delay fault. In such a case, the circuit may pass a test set containing traditional robust tests despite having a delay fault. Hence, even a small difference in τ-robust coverage might have a significant impact on delay test quality.

6.3 Practical Issues

For each target, the complexity of test generation is affected by two factors. The number of conditions that a τ-robust test must satisfy is typically much larger than that for a classical robust test for the corresponding path. Typically, this fact reduces the search space for test generation but also causes the τ-robust fault coverage to be low. However, in some cases there are many ways to switch off strings that are in parallel with on-target strings. These choices may increase the complexity of test generation.

Just like paths in circuits comprised of primitive gates, the number of targets may grow exponentially with an increase in the number of lines in a circuit. In fact, the number of targets can be much higher than the number of paths if a circuit has a large number of complex gates. However, most engineering simplifications proposed to make path delay testing practical can also be adapted to make τ-robust testing practical. First, algorithms may be developed to select a (small) subset of targets to be tested. Alternatively, the notion of segment delay testing may be adopted to consider *subtargets*. Secondly, if necessary, the stringent conditions for τ-robustness may be relaxed by adapting the notions of non-robust and other types of tests. Thirdly, techniques can be developed to identify (and hence eliminate from consideration) redundant and τ-robust dependent targets. In fact, removing functionally unsensitizable targets from consideration can be an effective way to reduce test generation time.

The circuit model presented in this paper ignores the effects of simultaneous switching of adjacent transistors on on-target strings, and of capacitances internal to gates. Note that the conditions for exciting effects mentioned in [2] and [4] do not run counter to the conditions proposed in this paper, and represent different dimensions of the worst-case delay excitation problem. The conditions for generating tests incorporating all the effects can be integrated into a single system to form a comprehensive framework for delay testing. An integrated approach would only result in a refinement of the individual set of conditions proposed for each of the above effects.

7 Conclusion and Future Work

Testing a circuit containing complex CMOS gates using path delay tests generated with a primitive gate-level model and classical robust test conditions can fail to excite worst-case delays in the circuit. A new entity for test, called a target, is proposed, conditions to ro-

Table 1. Comparison of τ-robust coverage on complex gate versions of ISCAS89 circuits.

Ckt.	CG	P	TP	τ	τ_LP	τ_FUS	τ-robust fault cov. of τ-robust test set T obtained by test generation				τ-robust fault cov. by fault simulation with a classical robust test set T'		$((T\tau(T) - T\tau(T'))/T\tau(T))$ (× 100 %)	$((T\tau\_NLP(T) - T\tau\_NLP(T'))/T\tau\_NLP(T)$ (× 100 %)
							$T\tau$ (T)	$T\tau\_LP$ (T)	$T\tau\_NLP$ (T)	$UT\tau$ (T)	$T\tau$ (T')	$T\tau\_NLP$ (T')		
$s208^*$	13	290	290	566	190	19	235	190	45	331	190	0	19.0	100.0
$s298^*$	13	462	343	647	327	121	331	262	69	316	296	34	10.6	50.7
$s344^*$	14	710	611	1614	496	459	562	434	128	1052	481	47	14.0	63.3
$s349^*$	14	730	611	1634	516	469	562	434	128	1072	481	47	14.0	63.3
$s386^*$	13	414	413	488	372	12	414	371	43	74	385	14	7.0	67.4
$s420^*$	13	738	738	2613	568	7	713	568	145	1900	568	0	20.0	100.0
$s444^*$	14	1070	586	1268	922	221	552	514	38	716	537	23	2.7	39.5
$s510^*$	17	738	729	839	664	3	738	656	82	101	675	19	8.5	76.8
$s526^*$	14	820	694	908	757	53	724	633	91	184	654	21	9.7	76.9
$s641^*$	14	3444	1941	5379	2180	751	2107	1346	761	3272	1844	498	12.5	34.6
$s820^*$	12	984	980	1045	934	8	969	930	39	76	952	22	1.8	43.6
$s832^*$	12	1012	984	1078	960	10	973	934	39	105	954	20	2.0	48.7
$s838^*$	11	2018	2018	4756	1644	0	1996	1644	352	2760	1670	26	16.0	92.6

bustly test a target are provided, and their completeness is proven (under the assumed delay model) to ensure complete delay fault testability of the circuit. Simulation results are given to demonstrate the effectiveness of the new methodology. τ-robust fault coverages of a τ-robust test set and a classical robust test set are compared for versions of ISCAS89 benchmark circuits with complex gates. It is shown that τ-robust tests are of significantly higher quality compared to the classical robust tests in terms of exciting the worst-case circuit delays, and hence their use can have a significant impact on test escapes.

The proposed theory can be extended in different directions. First, the fault model can be relaxed to consider a fault in a single transistor, a fault in a single complex gate or a fault in a single target. Second, weaker versions of tests must be developed for targets that are not τ-robustly testable. For example, conditions that can be specified for testing groups of targets that are not individually τ-robustly testable, and the guarantees associated with such tests in terms of the delays excited can be explored. Third, generation of compact τ-robust test sets must be investigated. Finally, this theory can be extended to include other types of logic, such as non-series-parallel CMOS gates, dynamic logic and pass transistor logic. Such new theory will enable trade-offs with respect to area, performance, and testability when performing technology mapping that involves selection of standard (possibly) complex CMOS cells from a library for the function to be implemented.

Acknowledgment

We thank Mohammed Saffat Quasem for helping in constructing complex gate versions of the benchmarks, and Nabil Abdulrazzaq for many useful discussions.

References

[1] S. Bose, V. D. Agrawal and T. G. Szymanski, "Algorithms for Switch Level Delay Fault Simulation," Proc. Int'l Test Conf., pp. 982-991, 1997.

[2] L-C. Chen, S. K. Gupta and M. A. Breuer, "High Quality Robust Tests for Path Delay Faults," Proc. IEEE VLSI Test Symp., pp. 88-93, 1997.

[3] R. I. Damper and N. Burgess, "MOS Test Pattern Generation Using Path Algebras," IEEE Tran. on Computers, vol. C-36, no. 9, pp. 1123-1128, September 1987.

[4] P. Franco and E. J. McCluskey, "Three Pattern Tests for Delay Faults," Proc. 12th IEEE VLSI Test Symp., pp. 452-456, May 1994.

[5] E. P. Hsieh, R. A. Rasmussen, L. J. Vidunas and W. T. Davis, "Delay Test Generation," Proc. 14th Design Automation Conf., pp. 486-491, June 1977.

[6] C. J. Lin and S. M. Reddy, "On Delay Fault Testing in Logic Circuits," IEEE Tran. on Computer Aided Design, vol. C-6, pp. 694-703, September 1987.

[7] S. Natarajan, S. K. Gupta and M. A. Breuer, "T-Robust Testing for Delay Faults," USC Technical Report CENG 98-06, 1998.

[8] J. Rajski and H. Cox, "Stuck-Open and Transition Fault Testing in CMOS Complex Gates," Proc. Int'l Test Conf., pp. 688-694, September 1988.

[9] M. K. Reddy, S. M. Reddy and P. Agrawal, "Transistor Level Test Generation for MOS Circuits," Proc. 22nd Design Automation Conf., pp. 825-828, 1985.

[10] N. K. Jha and S. Kundu, "Testing and Reliable Design of CMOS Circuits," Kluwer Academic Publishers, 1990.

[11] G. L. Smith, "Model for Delay Faults Based upon Paths," Proc. Int'l Test Conf., pp. 342-349, November 1985.