# Path delay fault testing of multiplexer-based shifters*

H. T. VERGOS†¶, Y. TSIATOUHAS‡, TH. HANIOTAKIS‖, D. NIKOLOS† and M. NICOLAIDIS§

In this paper we present a method for path delay fault testing of multiplexer-based shifters. We show that many paths of the shifter are not single path propagating hazard free robustly testable (SPP-HFRT) and we present a path selection method such that all the selected paths are SPP-HFRT by ($O \log_2 n$) test-vector pairs, where $n$ is the length of the shifter's operand. The propagation delay along all other paths is a function of the delays along the selected paths. This is the first work addressing the problem of shifter path delay fault testing.

## 1. Introduction

The increasing performance requirements of contemporary VLSI circuits make it difficult to design VLSI circuits that have large timing margins. Thus imprecise delay modelling, statistical variations of the parameters during the manufacturing process and physical defects in integrated circuits can all degrade circuit performance without altering the logic functionality. The change in the timing behaviour of the circuit is modelled by three common fault models. Firstly, the gross delay fault model (Brazilai and Rosen 1983) addresses delay defects that affect single lines in the circuit, causing the propagation delay through the lines to be 'very large'. Secondly, the gate delay fault model (Carter *et al.* 1987) also addresses defects that affect single lines, however, no assumption is made on the delay size. Finally, the path delay fault model (Smith 1985) addresses distributed or accumulated delays due to the propagation through several lines, each affected by a delay defect; therefore it is deemed to be more general. Two major problems are associated with path delay fault testing.

(a) The number of physical paths in a contemporary circuit is excessively large. Testing all of them for path delay faults is usually unaffordable.

(b) Since the single fault assumption is not realistic for the path delay fault model (a single defect usually will affect a large number of paths), a robust

Received 14 March 2000. Accepted 2 April 2001.

* A preliminary version of a part of this work entitled 'On Path Delay Fault Testing of Multiplexer-Based Shifters' by H. T. Vergos, Y. Tsiatouhas, Th. Haniotakis, D. Nikolos and M. Nikolaidis, appeared in the *9th Great Lakes Symposium on VLSI*, Ann Arbor, Michigan, March 4–6, 1999, pp. 20–23. © 1999 IEEE.

† Department of Computer Engineering and Informatics, University of Patras, 26 500, Rio, Patras, Greece.

‡ Integrated Systems Development S.A, 22, K. Varnali Str., 152 33 Maroussi, Athens, Greece.

‖ Department of Electrical Engineering, Southern Illinois University, 62901 Carbondale, USA.

§ Reliable Integrated Systems Group, TIMA, 46, Avenue, Felix Viallet, 38031, Grenoble, France.

¶ Corresponding author. e-mail: vergos@cti.gr

*International Journal of Electronics* ISSN 0020–7217 print/ISSN 1362–3060 online © 2001 Taylor & Francis Ltd
http://www.tandf.co.uk/journals
DOI: 10.1080/0020721011005813 9

test is usually required to detect a path delay fault. For many circuits, however, a large number of path delay faults is not robustly testable.

To reduce the number of paths that must be tested during manufacturing testing, assuming path delay faults, a number of path selection methods have been proposed. None of them though has proven to be satisfactory for the general case. Selecting all paths whose calculated delay exceeds a specific threshold is the simplest delay-based approach. However, the number of paths selected by this method is so large in general that all the selected paths cannot be tested. In the case of performance optimized circuits this is not a viable approach (Williams *et al.* 1991). The method presented in Li *et al.* (1989) selects a set of paths such that for each interconnect $l$ of a given circuit the set contains at least one path with the largest calculated delay among the paths through $l$. This method selects a number of paths that is moderate. The method, recently proposed in Tani *et al.* (1999), reduces the number of paths to be tested by judging which of two paths has the larger real delay. Both methods (Li *et al.* 1989, Tani *et al.* 1999) suffer from the problem that the selected paths may not be robustly testable (a large percentage of paths in a circuit is not robustly testable).

A number of functional approaches have also been proposed (Cheng and Chen 1993, Lam *et al.* 1993, Gharaybeh *et al.* 1995, Sparmann *et al.* 1995) as alternatives for the path selection problem. In these approaches, path selection is achieved by excluding paths that do not have to be tested functionally, such as unsensitizable paths and robust-dependent paths. However, these approaches are not practical for large circuits because the number of the selected paths is quite large.

Even if effective path selection is performed, the fault coverage attained by automatic test pattern generation on the selected paths is very hard to estimate, because this either requires an exponential number of paths to be investigated in exponential time complexity (Gharaybeh *et al.* 1996) or a non-enumerative method is used which does not provide completely accurate results (Pomeranz and Reddy 1994).

Another completely different approach is the use of test sets derived for stuck-at-faults (properly modified for transition faults) for path delay fault testing, as proposed in Pomeranz and Reddy (2000). Although the required test vectors can be derived easily for this approach, a high path delay fault coverage is not guaranteed.

Lesser and Shedletsky (1980) have shown that by measuring the propagation delays along a subset of paths (called a *basis*), the propagation delay along every other path of the circuit can be calculated in a straightforward method, leading to a path delay fault coverage of 100%. However, this method has not yet found wide application. We believe that the main reason is that its application requires the direct accessibility of the inputs and the outputs of the circuit under test. Nowadays a circuit is often embedded in a larger module, therefore its inputs and outputs are not directly accessible. The problem of the accessibility was recently overcome by the method proposed in Nikolos *et al.* (1999). There are also two more reasons that have discouraged the application of the method given by Lesser and Shedletsky (1980).

(a) In order to measure the propagation delays of the paths of the basis during path delay fault testing the measurement should not be invalidated by possible delay faults or hazards along other paths of the circuit. In other words there is the requirement that the paths of the basis are single path propagating hazard free robustly testable (SPP-HFRT). Although for several circuits a basis consisting of SPP-HFRT vectors does not exist, there are circuits with such a basis (see for example Haniotakis *et al.* (1998), Bellos *et al.* (1999) and Vergos *et al.* (1999)) or circuits that can be slightly modified so that such a basis can be derived (for example, Haniotakis *et al.* (1999), Kalligeros *et al.* (1999) and Sidiropoulos *et al.* (1999)).

(b) Based on the method given in Lesser and Shedletsky (1980) an exhaustive search is required, in the general case, to find a basis consisting of only SPP-HFRT paths if such a basis exists for the circuit under consideration. Moreover, the method given in Lesser and Shedletsky (1980) cannot exploit the possible inherent parallelism of a circuit. However, as we shall show in this paper, taking into account the structural properties of some circuits the exhaustive search can be avoided.

This work addresses the problem of shifters' path delay fault testing and is a part of a broader project concerning the path delay fault testing of all modules included in a data path (for example adders (Haniotakis *et al.* 1998) and multipliers (Haniotakis *et al.* 1999, Kalligeros *et al.* 1999)). Data paths are essential and generally the biggest logic parts of microprocessors and microcontrollers. Shifters can be implemented in different formats, such as barrel-shifters or multiplexer-based (Weste and Eshraghian 1993). This work considers the latter implementations since standard cell designs are becoming predominant in industrial context and can be easily automated through the use of a hardware description language (HDL). We assume gate level implementations for the shifters. The inputs and outputs of the shifter are, according to Nikolos *et al.* (1999), treated in the following as primary inputs and primary outputs.

In § 2 we give the definition of single path propagating hazard free robustly testable (SPP-HFRT) paths and the design of the shifters under consideration. A new systematic path selection method is presented in § 3 and we also prove that the selected paths are SPP-HFRT and that they constitute a basis for the shifter. In § 4, after deriving the number of paths of the shifter, we present comparison results indicating the test effort reduction achieved by the proposed path selection method.

## 2. Preliminaries

A two pattern test $T = \langle V_1, V_2 \rangle$ is said to be a robust delay test for a path $P$, for a rising or falling transition at the output of the path, if and only if, when $P$ is faulty and test $T$ is applied, the circuit output is different from the expected state at sampling time, independent of the delays along gate inputs not on $P$ (Lin and Reddy 1987). A robust test that propagates the fault effect through only a single path to an output of the circuit will be called a single-path propagating robust test (SPP-RT) for that output (Prammanick and Reddy 1990). A robust test is said to be a hazard-free robust test (HFRT) if no hazards can occur on the tested path during the application of the test, regardless of the gate delay values. Robust tests may not exist for all path delay faults in an arbitrary circuit.

The method given in Lesser and Shedletsky (1980) is based on the fact that common segments of two logical paths have exactly the same delay on both paths. In general the delay along a path depends on the specific values of the off-inputs of the path. (An input is an off-input of path $P$ if it is an input to a gate in $P$ but does not belong to the path.) The reason is that when we measure the delay along the two paths the off-inputs of the common segment may be in different values, for example at 0 and 1 respectively for an exclusive-OR gate. However, in the case that the circuit

| $t_1 t_0$ | Operation |
|---|---|
| 00 | Rotate right |
| 01 | Logical left shift |
| 10 | Logical right shift |
| 11 | Arithmetic right shift |

Table 1. Shifter's functionality

consists only of AND, NAND, OR, NOR and NOT gates all the off-inputs of a gate will be at the same non-controlling value, for example at 1 for the AND gates and at 0 for the OR gates. Note that the multiplexer-based shifters are considered to be implemented by AND and OR gates, therefore the method given in Lesser and Shedletsky (1980), can be applied.

In the majority of cases, the length $n$ of the shifter's operand is a power of 2, that is $n = 8, 16, 32, 64, \ldots$. Therefore, we consider $n = 2^m$, with $m = 1, 2, 3, \ldots$. An $n$-bit multiplexer-based shifter capable of performing up to $n - 1$ positions shift of its input operand in a single clock cycle is a combinational circuit made up of $m = \log_2 n$ levels. Each level requires $n$ building blocks and is capable of performing a shift function of $2^i$ positions according to the value of $C_i$, $i = 0, 1, \ldots, m - 1$ and $C_i \in \{0, 1\}$, which is the common control signal of each level. Each level accepts as inputs the outputs of the previous level (or the primary inputs) and only drives the subsequent level (or the primary outputs). A value of 0 at $C_i$ indicates that no shifting will take place at the $i$th level of the shifter. Each possible shift function is selected according to the values of $t_1 t_0$ signals as shown in table 1. Every building block of the shifter accepts the same $t_1 t_0$ signal values, thus it performs the same shift function every clock cycle.

Figure 1 presents a 16-bit multiplexer-based shifter capable of performing the four different functions indicated in table 1. The connections between the various levels have been omitted for clarity. Lines that should be connected between the different levels have been named in a unique way. The basic building block of the shifter is composed of two multiplexers.

(a) A 4 → 1 multiplexer controlled by the $t_1 t_0$ signals which implements the one out of four possible shift functions. When $t_1 t_0$ have the 00 or 01 or 10 or 11 value the rightmost, the next to the rightmost, the next to the leftmost and the leftmost input of every 4 → 1 MUX are driven respectively to its output.

(b) A 2 → 1 multiplexer controlled by the appropriate $C_i$ signal.

## 3. Path selection

During the normal operation of a multiplexer-based shifter, there are transitions at the data inputs, the control signals $C_{m-1} C_{m-2} \ldots C_1 C_0$, as well as the function signals $t_1 t_0$. This means that for path delay fault testing of the shifter, we must consider delay faults along paths that are driven from any of these possible sources.

A physical path $p$ of a circuit is a route through gates and interconnection lines leading from a primary input $p_{\text{in}}$ to a primary output $p_{\text{out}}$ of the circuit. In delay fault test generation we associate two logical paths with each physical path $p$ of the circuit. A logical path is a pair $(T, p)$, where $T$ is a transition from $\bar{x}$ to $x$ at $p_{\text{in}}$, with $x \in \{0, 1\}$ and $\bar{x}$ the complement of $x$. In the case of delay fault testing the test set

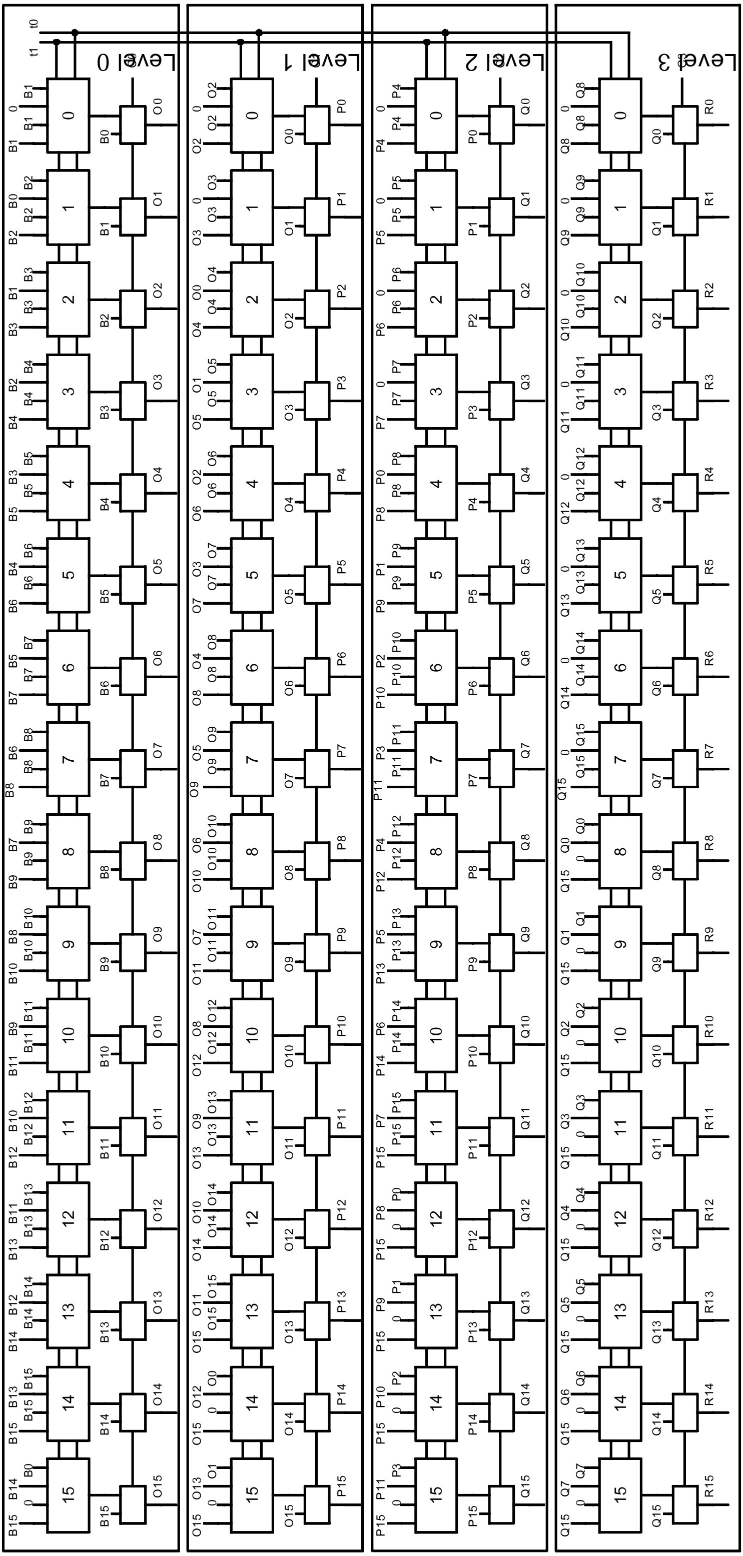

Figure 1. A 16-bit multiplexer-based shifter capable of performing four different logic functions.

consists of pairs of vectors. We define a *test session* as the application of a pair of test vectors which sensitize a certain path $p$ and propagate a transition ($0 \rightarrow 1$ or $1 \rightarrow 0$) from $p_{\text{in}}$ to $p_{\text{out}}$. We note that in a completely robustly testable circuit for each transition along every path a pair of test vectors can always be found. If during a test session more than one distinct path can be sensitised in parallel and can propagate transitions to distinct primary outputs, then all these paths can be tested in parallel for delay faults, thus reducing testing effort and the test application time significantly.

In a multiplexer-based shifter, we divide all the possible physical paths in the following three categories.

### 3.1. *Paths starting from the data inputs*

We define the paths established by a specific combination of the control signals $C_{m-1}C_{m-2}\ldots C_1C_0$ and the function signals $t_1t_0$ as a group of paths or simply a group.

**Lemma 1:** *The propagation delays along all the paths of a group can be measured in parallel.*

**Proof:**

(a) The paths of each group defined by $t_1t_0 \in \{00, 01, 10\}$ and any value of the control signals $C_{m-1}C_{m-2}\ldots C_1C_0$ do not have any common sub-path. Thus the propagation delays along these paths can be measured in parallel.

(b) For the physical paths of each group defined by $t_1t_0 = 11$ and any value of the control signals $C_{m-1}C_{m-2}\ldots C_1C_0$ we can observe that a transition starting from an input, after propagating along a sub-path may then propagate through several sub-paths to an output of the shifter. Since the latter sub-paths do not re-converge (specifically each sub-path ends to a distinct primary output of the shifter), the propagation delays along the paths of each one of these groups can also be measured in parallel. □

We define as *weight w* of a group the number of ones in the combination of the control signals $C_{m-1}C_{m-2}\ldots C_1C_0$ that establishes the paths of this group. According to this definition it is evident that the groups with same $C_{m-1}C_{m-2}\ldots C_1C_0$ value and different $t_1t_0$ values have the same weight.

The notation $B$-$O$ is used in the following to denote a path or a sub-path with input $B$ and output $O$. Also the notation $B$-$O/O$-$P$ is used to denote a path $B$-$P$ when its sub-path $B$-$O$ drives its sub-path $O$-$P$.

**Theorem 1:** *For a specific value of $t_1t_0$ the propagation delay along any path $P_1$ in a group with weight $w_1 > 1$ can be calculated by the propagation delays along a path $P_2$ in a group with weight $w_2 = w_1 - 1$, a path $P_3$ in a group with weight $w_3 = 1$ and a path $P_4$ in a group with weight $w_4 = 0$.*

**Proof:** Let $P_1$ be a path with $w_1 > 1$ and $P_1 = P_{1a}/P_{1b}$ where $P_{1a}$ is a sub-path from the data input to the output of the first level $i$ for which the corresponding control signal $C_i$ is 1. Let $P_4 = P_{4a}/P_{4b}$, be the path belonging to the group with weight 0, with $P_4 = P_{4a}/P_{4b}$ such that $P_{4a}$ and $P_{1a}$ end at the same point. Consider the paths $P_2 = P_{2a}/P_{2b}$ where $P_{2a} = P_{4a}$ and $P_{2b} = P_{1b}$, which has a weight $w_2 = w_1 - 1$ and $P_3 = P_{3a}/P_{3b}$ where $P_{3a} = P_{1a}$ and $P_{3b} = P_{4b}$, which has a weight of $w_3 = 1$. Then the propagation delay $d(P_1)$ along path $P_1$ can be expressed as a linear combination of the propagation delays $d(P_2)$, $d(P_3)$ and $d(P_4)$ along paths $P_2$, $P_3$ and $P_4$, respectively as: $d(P_1) = d(P_2) + d(P_3) - d(P_4)$. □

**Example 1:** For the shifter of figure 1 consider $t_1 t_0 = 00$. For the path $P_1 = B_{12}\text{-}O_{12}/O_{12}\text{-}P_{10}/P_{10}\text{-}Q_6/Q_6\text{-}R_6$, established by $C_3C_2C_1C_0 = 0110$, that is, $w_1 = 2$, with $P_{1a} = B_{12}\text{-}O_{12}/O_{12}\text{-}P_{10}$ and $P_{1b} = P_{10}\text{-}Q_6/Q_6\text{-}R_6$, consider the paths:

(a) $P_4 = B_{10}\text{-}O_{10}/O_{10}\text{-}P_{10}/P_{10}\text{-}Q_{10}/Q_{10}\text{-}R_{10}$, established by $C_3C_2C_1C_0 = 0000$, that is, $w_4 = 0$, with $P_{4a} = B_{10} - O_{10}/O_{10} - P_{10}$ and $P_{4b} = P_{10}\text{-}Q_{10}/Q_{10}\text{-}R_{10}$. Note that $P_{4a}$ and $P_{1a}$ end at the same point.

(b) $P_2 = B_{10}\text{-}O_{10}/O_{10}\text{-}P_{10}/P_{10}\text{-}Q_6/Q_6\text{-}R_6$, established by $C_3C_2C_1C_0 = 0100$, that is, $w_2 = w_1 - 1 = 1$, with $P_{2a} = P_{4a} = B_{10}\text{-}O_{10}/O_{10}\text{-}P_{10}$, $P_{2b} = P_{1b} = P_{10}\text{-}Q_6/Q_6\text{-}R_6$ and

(c) $P_3 = B_{12}\text{-}O_{12}/O_{12}\text{-}P_{10}/P_{10}\text{-}Q_{10}/Q_{10}\text{-}R_{10}$, established by $C_3C_2C_1C_0 = 0010$, that is, $w_2 = 1$, with $P_{3a} = P_{1a} = B_{12}\text{-}O_{12}/O_{12}\text{-}P_{10}$ and $P_{3b} = P_{4b} = P_{10}\text{-}Q_{10}/Q_{10}\text{-}R_{10}$.

We can easily see that the propagation delay along path $P_1$ can be expressed as: $d(P_1) = d(P_2) + d(P_3) - d(P_4)$.

From Theorem 1 the propagation delay along any path with weight $w$ greater than 1 can be calculated from the propagation delays along a path with weight $w - 1$, a path with weight 1 and a path with weight 0. The propagation delay along the path with weight $w - 1$ can be calculated from the propagation delays along a path with weight $w - 2$, a path with weight 1 and a path with weight 0. By induction we conclude that if the propagation delays along all paths in groups with

| $C_3C_2C_1C_0$ | $t_1t_0$ | $B_{15}B_{14}\ldots B_0$ | Weight |
|---|---|---|---|
| 0000 | 00 or 01 or 10 or 11 | (TT…T), (˜T˜T…˜T) | 0 |
| 0001 | 00 | (TT…T), (˜T˜T…˜T) | 1 |
| 0010 | 00 | (TT…T), (˜T˜T…˜T) | 1 |
| 0100 | 00 | (TT…T), (˜T˜T…˜T) | 1 |
| 1000 | 00 | (TT…T), (˜T˜T…˜T) | 1 |
| 0001 | 01 | (TT…T), (˜T˜T…˜T) | 1 |
| 0010 | 01 | (TT…T), (˜T˜T…˜T) | 1 |
| 0100 | 01 | (TT…T), (˜T˜T…˜T) | 1 |
| 1000 | 01 | (TT…T), (˜T˜T…˜T) | 1 |
| 0001 | 10 | (TT…T), (˜T˜T…˜T) | 1 |
| 0010 | 10 | (TT…T), (˜T˜T…˜T) | 1 |
| 0100 | 10 | (TT…T), (˜T˜T…˜T) | 1 |
| 1000 | 10 | (TT…T), (˜T˜T…˜T) | 1 |
| 0001 | 11 | (TT…T), (˜T˜T…˜T) | 1 |
| 0010 | 11 | (TT…T), (˜T˜T…˜T) | 1 |
| 0100 | 11 | (TT…T), (˜T˜T…˜T) | 1 |
| 1000 | 11 | (TT…T), (˜T˜T…˜T) | 1 |

T denotes a 0 → 1 or a 1 → 0 transition. ˜T denotes the inverse transition.

Table 2. Test sessions required for delay fault testing of the paths starting from the data inputs for the shifter of figure 1.

weight 0 and 1 are known, then we can calculate the propagation delays along any path with greater weight. This means that during testing we only need to measure the propagation delays along all paths in groups with weight 0 and 1 for every $t_1 t_0$ value. Moreover, the groups of paths with weight 0 are identical for any possible value of $t_1 t_0$. Thus, the propagation delays along these paths need only be measured once.

The number of test sessions needed to measure the propagation delays along the paths of groups with weight 0 and 1 is equal to: $2 * (4 * \log_2 n + 1)$, where $n$ is the length of the shifter's operand and $*$ denotes multiplication. For the shifter of figure 1 the number of such test sessions is 34. Table 2 indicates these test sessions.

### 3.2. *Paths starting from the control inputs*

A logical path from a control signal $C_i$ through a $2 \rightarrow 1$ MUX, for which $C_i$ is the control input, is established iff the inputs of the multiplexer have complementary values. In other words, if the inputs of the $2 \rightarrow 1$ multiplexer are not at complementary values then a transition on the control input $C_i$ cannot propagate to the output of the $2 \rightarrow 1$ multiplexer. Obviously there are four such logical paths from a control signal $C_i$ through a $2 \rightarrow 1$ MUX for which $C_i$ is the control input, the two logical paths with the inputs of the multiplexer having the values 0 and 1 (figure 2(*a*)) and the two with the inputs of the multiplexer having the values 1 and 0 (figure 2(*b*)).

Consider now a control signal input $C_i$, and also that for every $C_j$ with $j < i$, $C_j = 0$. Furthermore, consider that for the function signals we have the values $t_1 t_0 = 00$. In order to achieve that the inputs of all $2 \rightarrow 1$ multiplexers at level $i$ have complementary values, we divide the data inputs in two sets. The first set is constructed from the data inputs numbered $x$, where $x \bmod 2^{i+1} < 2^i$, and the second set is constructed from the data inputs numbered $y$, where $y \bmod 2^{i+1} \geq 2^i$. All the data inputs of each set are driven with the same value. The two sets are assigned with complementary values.

**Example 2:** Consider the control signal input $C_2$ in figure 1. In order to achieve that the inputs of all $2 \rightarrow 1$ multiplexers at level 2, have complementary values we divide the data operand inputs in two sets, one set with the inputs $\{B_0, B_1, B_2, B_3, B_8, B_9, B_{10}, B_{11}\}$ and another set with the inputs $\{B_4, B_5, B_6, B_7, B_{12}, B_{13}, B_{14}, B_{15}\}$. Complementary values should be assigned to these two sets of inputs. Furthermore, we must set $C_1 C_0 = 00$ and $t_1 t_0 = 00$. To measure the propagation delay for the transition of $C_2$: $0 \rightarrow 1$ and $1 \rightarrow 0$ we have to set $B_0 = B_1 = B_2 = B_3 = B_8 = B_9 = B_{10} = B_{11} = 0$ and $B_4 = B_5 = B_6 = B_7 =$

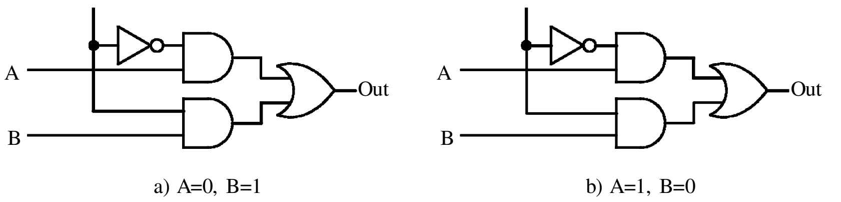

Figure 2. The two paths starting from the control signal of a multiplexer for complementary input values. Each thick line represents two logical paths.

$B_{12} = B_{13} = B_{14} = B_{15} = 1$ and then $B_0 = B_1 = B_2 = B_3 = B_8 = B_9 = B_{10} = B_{11} = 1$ and $B_4 = B_5 = B_6 = B_7 = B_{12} = B_{13} = B_{14} = B_{15} = 0$ for each of the transitions

Let $L_i$ be the set of all paths starting from a control signal $C_i$ with all other control signals $C_j = 0$, for $j \neq i$ and $t_1 t_0 = 00$.

**Lemma 2:** *The propagation delays along all paths of $L_i$, can be measured in parallel, for every one of the two combinations of complementary values for the two groups of data inputs.*

**Proof:** Since $C_j = 0$ for every $j \neq i$ all these paths do not have any common sub-path, thus the propagation delays along them can be measured in parallel. □

For every level $i$ the number of test sessions required for measuring the propagation delays along the paths in set $L_i$ is 4. Thus in order to measure the propagation delay times along the paths in all sets $L_i$, with $0 \leq i \leq m - 1$, we need $4 * m = 4 * \log_2 n$ test sessions. Table 3 indicates the 16 test sessions for the shifter of figure 1.

**Theorem 2:** *If we measure the propagation delays along all the paths in set $L_i$, the propagation delay along every other path starting form $C_i$ can be calculated using also the propagation delays along paths starting from the data inputs.*

**Proof:** Suppose $P_1 = P_{1a}/P_{1b}$ is a path starting from $C_i$ and $P_1 \notin L_i$. This means that $\exists j > i$ such that $C_j = 1$. Let $P_{1a}$ be the sub-path from $C_i$ to the output of the $2 \rightarrow 1$ multiplexer of level $i$. Consider the following paths:

(a) $P_2 = P_{2a}/P_{2b}$ of $L_i$ with $P_{2a} = P_{1a}$,

| $B_{15}B_{14}B_{13}B_{12}B_{11}B_{10}B_9B_8B_7B_6B_5B_4B_3B_2B_1B_0$ | $C_3C_2C_1C_0$ |
|---|---|
| 1010101010101010 | 000T |
| 0101010101010101 | 000T |
| 1010101010101010 | 000~T |
| 0101010101010101 | 000~T |
| 1100110011001100 | 00T0 |
| 0011001100110011 | 00T0 |
| 1100110011001100 | 00~T0 |
| 0011001100110011 | 00~T0 |
| 1111000011110000 | 0T00 |
| 0000111100001111 | 0T00 |
| 1111000011110000 | 0~T00 |
| 0000111100001111 | 0~T00 |
| 1111111100000000 | T000 |
| 0000000011111111 | T000 |
| 1111111100000000 | T~000 |
| 0000000011111111 | T~000 |

During the application of the above $t_1 t_0 = 00$.

Table 3. Test sessions required for delay fault testing of the paths starting from the control inputs for the shifter of figure 1.

(b) the path $P_3 = P_{3a}/P_{3b}$, starting from a data input (case 3.1), where $P_{3a}$ ends at the same with $P_{1a}$ $2 \rightarrow 1$ multiplexer output of level $i$ with $C_j = 0$ for $j < i$ and $P_{3b} = P_{1b}$, and

(c) the path $P_4 = P_{4a}/P_{4b}$, starting from a data input (case 3.1), with $P_{4a} = P_{3a}$ and $P_{4b} = P_{2b}$. Note that $P_4$ belongs to the group of weight 0.

Then the propagation delay $d(P_1)$ along path $P_1$ can be expressed as: $d(P_1) = d(P_2) + d(P_3) - d(P_4)$ where $d(P_2)$, $d(P_3)$ and $d(P_4)$ represent the propagation delays along paths $P_2$, $P_3$ and $P_4$, respectively.

Note that the paths $P_3$ and $P_4$ are established for the same values of $t_1 t_0$ as $P_1$ does but since $P_2 \in L_i$ the propagation delay along path $P_2$ has only been measured with $t_1 t_0 = 00$. Since the propagation delay along any path or sub-path does not depend on the values of $t_1 t_0$, this does not invalidate the proof. □

**Example 3:** For the shifter of figure 1 consider that we wish to calculate the propagation delay along the path $P_1 = C_2\text{-}Q_0/Q_0\text{-}R_8$, established when $t_1 t_0 = 01$ and $C_3 C_1 C_0 = 100$. Assume that $P_1 = P_{1a}/P_{1b}$, with $P_{1a} = C_2\text{-}Q_0$ and $P_{1b} = Q_0\text{-}R_8$.

(a) The propagation delay along the path $P_2 = P_{2a}/P_{2b} = C_2\text{-}Q_0/Q_0\text{-}R_0$, established when $t_1 t_0 = \mathrm{XX}$ (X represent don't care conditions) and $C_3 C_1 C_0 = 000$, has been measured since $P_2 \in L_2$ with $t_1 t_0 = 00$. Note that $P_{2a} = P_{1a} = C_2\text{-}Q_0$.

(b) The propagation delay along the path: $P_3 = B_0\text{-}O_0/O_0\text{-}\ P_0/P_0\text{-}Q_0/Q_0\text{-}R_8$, established by $C_3 C_2 C_1 C_0 = 1000$ and $t_1 t_0 = 01$ has been measured (case 3.1). Note that $P_3 = P_{3a}/P_{3b}$, with $P_{3a} = B_0\text{-}O_0/O_0\text{-}P_0/P_0\text{-}Q_0$ and $P_{3b} = P_{1b} = Q_0\text{-}R_8$ and that $P_{3a}$ ends at the same point as $P_{1a}$.

(c) The propagation delay along $P_4 = B_0\text{-}O_0/O_0\text{-}P_0/P_0\text{-}Q_0/Q_0\text{-}R_0$, established by $C_3 C_2 C_1 C_0 = 0000$ and $t_1 t_0 = \mathrm{XX}$ has been measured (case 3.1). Note that $P_4 = P_{4a}/P_{4b}$ with $P_{4a} = P_{3a} = B_0\text{-}O_0/O_0\text{-}P_0/P_0\text{-}Q_0$ and $P_{4b} = P_{2b} = Q_0\text{-}R_0$ belongs to the group with 0 weight.

Then $d(P_1) = d(P_2) + d(P_3) - d(P_4)$

### 3.3. *Paths starting from the function inputs*

In this case we consider delay fault testing along paths with input either $t_0$ or $t_1$ and output one of the primary outputs of the shifter for constant values at the data inputs and $C_{m-1} C_{m-2} \ldots C_0$.

Since $t_0$ and $t_1$ drive all levels of the shifter these paths are not all SPP-HFRT. For example consider the path with input $t_0$ and output $R_3$ in figure 1 for $C_0 = C_1 = 1$, $C_2 = C_3 = 0$, $B_0 = 0$, $B_2 = 1$ and $t_1 = 0$. This path is not SPP-HFRT because the AND gate in the $4 \rightarrow 1$ multiplexer with label 3 in level 1, that belongs on the path is driven also by $t_0$ (see figure 3). The same is valid for the paths where more than one of the signals $C_{m-1} C_{m-2} \ldots C_0$ are equal to 1.

Let $M_i$ and $Q_i$ be the sets of all paths starting from $t_0$ and $t_1$ respectively with $C_i = 1$ and all the other control signals $C_j = 0$, with $j \neq i$. The measurement of the propagation delays along the paths of $M_i$ and $Q_i$ for $i = 0, 1, 2, \ldots \log_2 n - 1$ can be done as the paths of $L_i$ in case 3.2. Specifically the data inputs should be set to both the values:

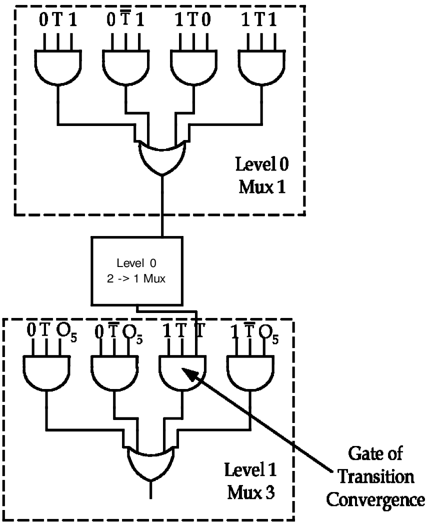

Figure 3. Re-convergence of transitions of $t_0$.

(i) $2^i$ zeroes, $2^{i+1}$ ones, $2^{i+1}$ zeroes, $2^{i+1}$ ones... and

(ii) $2^i$ ones, $2^{i+1}$ zeroes, $2^{i+1}$ ones, $2^{i+1}$ zeroes...

for each transition $0 \rightarrow 1$ and $1 \rightarrow 0$ of $t_0$ or $t_1$, in order to measure the propagation delays of all paths belonging to $M_i$ and $Q_i$.

For every level $i$ the number of test sessions in order to measure the propagation delays along the paths in sets $M_i$ and $Q_i$ depends on the implementation of the $4 \rightarrow 1$ multiplexer. If the multiplexer is designed as shown in figure 3 the required number of test sessions is 16. If it is designed as a tree of $2 \rightarrow 1$ multiplexers the required number of test sessions is 12. Without loss of generality, in the following we assume that the $4 \rightarrow 1$ multiplexers are designed as in figure 3. Thus in order to measure the propagation delay times along the paths in all sets $M_i$ and $Q_i$, with $0 \leq i \leq m-1$, we need $16 * \log_2 n$ test sessions.

**Theorem 3:** *If we measure the propagation delays along all the paths in sets $M_i$ and $Q_i$, the propagation delay along every other path starting from either $t_1$ or $t_0$ can be calculated using also the propagation delays along paths starting from the data inputs.*

**Proof:** Suppose $P_1 = P_{1a}/P_{1b}$ is a path starting from $t_0$ and $P_1 \notin M_i$. Let $P_{1a}$ be the sub-path from $t_0$ to the output of the $2 \rightarrow 1$ multiplexer of level $i$, with $i$ denoting the minimum value of $C_0 C_1 \ldots C_{m-1}$, such that $C_i = 1$. Consider the following paths:

(a) $P_2 = P_{2a}/P_{2b}$ of $M_i$ with $P_{2a} = P_{1a}$,

(b) $P_3 = P_{3a}/P_{3b}$, where $P_{3a}$ starts from a data input and ends to the same $2 \rightarrow 1$ multiplexer output as $P_{1a}$, and $P_{3b} = P_{1b}$ (case 3.1) and

(c) $P_4 = P_{4a}/P_{4b}$, where $P_{4a} = P_{3a}$ and $P_{4b} = P_{2b}$ (case 3.1). Obviously, $P_4$ belongs to the group of weight 0.

Then the propagation delay $d(P_1)$ along path $P_1$ can be expressed as: $d(P_1) = d(P_2) + d(P_3) - d(P_4)$ where $d(P_2)$, $d(P_3)$ and $d(P_4)$ represent the propagation delays along paths $P_2$, $P_3$ and $P_4$, respectively.

The same procedure can be followed for the proof concerning the paths starting from $t_1$. □

**Example 4:** For the shifter of figure 1 consider the path $P_1 = t_0\text{-}O_4/O_4\text{-}P_2/P_2\text{-}Q_2/Q_2\text{-}R_2$, established by $C_0 = C_1 = 1$, $C_2 = C_3 = 0$ and $t_1 = 0$. Assume that $P_1 = P_{1a}/P_{1b}$ with $P_{1a} = t_0\text{-}O_4$ and $P_{1b} = O_4\text{-}P_2/P_2\text{-}Q_2/Q_2\text{-}R_2$. As explained above this path is not SPP-HFRT. Consider the following paths:

(a) $P_2 = t_0\text{-}O_4/O_4\text{-}P_4/P_4\text{-}Q_4/Q_4\text{-}R_4$, of $M_0$ established by $C_0 = 1$, $C_1 = C_2 = C_3 = 0$ and $t_1 = 0$. Note that $P_2 = P_{2a}/P_{2b}$, with $P_{2a} = P_{1a} = t_0\text{-}O_4$ and $P_{2b} = O_4\text{-}P_4/P_4\text{-}Q_4/Q_4\text{-}R_4$.

(b) $P_3 = P_{3a}/P_{3b} = B_5\text{-}O_4/O_4\text{-}P_2/P_2\text{-}Q_2/Q_2\text{-}R_2$, established by $C_0 = C_1 = 1$, $C_2 = C_3 = 0$, $t_1 = t_0 = 0$, with $P_{3b} = P_{1b} = O_4\text{-}P_2/P_2\text{-}Q_2/Q_2\text{-}R_2$. Path $P_3$ has a weight of 2 and therefore its propagation delay was calculated after the application of the vectors of table 2.

(c) $P_4 = P_{4a}/P_{4b} = B_5\text{-}O_4/O_4\text{-}P_4/P_4\text{-}Q_4/Q_4\text{-}R_4$ established by $C_0 = 1$, $C_1 = C_2 = C_3 = 0$, $t_1 = t_0 = 0$, with $P_{4a} = P_{3a} = B_5\text{-}O_4$ and $P_{4b} = P_{2b} = O_4\text{-}P_4/P_4\text{-}Q_4/Q_4\text{-}R_4$. Path $P_4$ has a weight of 1 and its propagation delay was measured during the application of the vectors of table 2.

Then we can easily see that $d(P_1) = d(P_2) + d(P_3) - d(P_4)$.

Summarizing the above analysis, the total number of test sessions that are required for path delay fault testing of an $n$-bit multiplexer-based shifter capable of performing the four functions indicated in table 1 by the proposed method is $28 * \log_2 n + 2$, assuming the 4 → 1 multiplexer implementation presented in figure 3.

## 4. Discussion

In this section we will show that the path selection method proposed in the previous section leads to an extremely small number of selected paths, with respect to all possible logical paths. To this end we derive the number of physical as well as logical paths of an $n$-bit shifter.

### 4.1. *Number of physical paths starting from the data inputs*

(a) When the function inputs are both zero for every control input combination $n$ distinct paths are established. Then for the $n$ possible control input combinations there are $n^2$ physical paths.

(b) When the function inputs receive either the 01 or the 10 value, for each control input combination excluding the all 0s (that is covered by case (a) above), there are $n - i$ physical paths, with $i$ denoting the paths that receive the constant 0 value. Then the number of physical paths is $2 * \sum_{i=1}^{n-1} n - i = n^2 - n$.

| Operand length $n$ | No. of logical paths | Test sessions required by the proposed method | Reduction % |
|---|---|---|---|
| 8 | 1 312 | 86 | 93.45 |
| 16 | 6 208 | 114 | 98.16 |
| 32 | 27 264 | 142 | 99.48 |
| 64 | 114 944 | 170 | 99.85 |

Table 4. Comparisons.

(c) When the function inputs are both 1, for every possible control input combination, excluding the all 0s, $n$ distinct paths are established. In this case the physical paths are $(n-1)*n = n^2 - n$.

Summing the above there are

$$3n^2 - 2n \tag{1}$$

physical paths starting from the data inputs.

### 4.2. *Number of physical paths starting from the control inputs*

Each control input $j$, with $0 \leq j < m$, drives $n$ $2 \rightarrow 1$ multiplexers and there are two physical paths from the control input of each multiplexer to its output. The rest control inputs can receive $2^{m-1}$ distinct values ($m = \log_2 n$). Of these values distinct paths are established according to the values only of the control inputs $j+1$, $j+2, \ldots, m-1$. Moreover all these paths are distinct for each value of the function control inputs. The above analysis leads to a number of physical paths equal to

$$\sum_{j=0}^{m-1} 2*n*4*2^{m-(j+1)} = 8*n\sum_{j=1}^{m} 2^{m-j} \tag{2}$$

### 4.3. *Number of physical paths starting from the function inputs*

Assuming the implementation of the $4 \rightarrow 1$ multiplexer presented in figure 3, there are four distinct paths connecting each of the two function inputs to the output of each $4 \rightarrow 1$ multiplexers. For each such output the combination of the control inputs of the subsequent levels excluding the all 0s combination establishes distinct paths to the outputs of the shifter. The above analysis leads to a number of physical paths equal to

$$\sum_{j=0}^{m-1} 2*n*4*(2^{m-j}-1) = 8*n\sum_{j=1}^{m}(2^{m-(j+1)}-1) \tag{3}$$

Summing the results of relations (1), (2) and (3), we get that the number of physical paths (NPP) is equal to: $\text{NPP} = 15n^2 - 14n - 8n\log_2 n$. The number of logical paths (NLP) is equal to $\text{NLP} = 2*\text{NPP}$.

We remind that for testing a logical path, one test session is required. Table 4 presents comparisons between the NLP and the number of test sessions required for the paths selected by the proposed method for various values of the length of the shifter's operand. From this table it is obvious that the proposed path selection procedure can lead to an extremely large reduction of the test effort required.

## 5. Conclusions

In this paper, we have proposed a new path selection procedure for path delay fault testing of multiplexer-based shifters. Although the number of all logical paths (physical paths $*2$) of the shifter is $O(n^2)$, where $n$ is the length of the shifter and many of them, as we have shown, are not robustly testable, under the proposed selection method the selected paths are robustly testable by $O(\log_2 n)$ test-vector pairs. We have also shown that the propagation delay along all the rest paths can be calculated from the delays along the selected paths. Therefore the application of the proposed method reduces significantly the test application time.

## References

Bellos, M., Nikolos, D., and Vergos, H. T., 1999, Path delay fault testing of a class of circuit-switched multistage interconnection networks. *Lecture Notes in Computer Science No. 1667.* J. Hlavicka, E. Maehle and A. Pataricza (Eds), *Proceedings of Third European Dependable Computing Conference, EDCC-3,* pp. 267–282 (Springer-Verlag).

Brazilai, Z., and Rosen, B., 1983, Comparison of AC self-testing procedures. *Proceedings of the International Test Conference,* pp. 89–94.

Carter, J. L., Iyengar, V. S., and Rosen, B., 1987, Efficient test coverage determination for delay faults. *Proceedings of International Test Conference* Philadelphia, PA (IEEE Computer Society Press), pp. 418–427.

Cheng, K. T., and Chen, H. C., 1993, Delay testing for non-robust untestable circuits. *Proceedings of the International Test Conference,* Baltimore, MD (IEEE), pp. 954–961.

Gharaybeh, M. A., Bushnell, M. L., and Agrawal, V. D., 1995, Classification and test generation for path-delay faults using single stuck-fault tests. *Proceedings of the International Test Conference,* Washington, DC (IEEE), pp. 139–148.

Gharaybeh, M. A., Bushnell, M. L., and Agrawal, V. D., 1996, An exact non-enumerative fault simulator for path delay faults. *Proceedings of the International Test Conference,* Washington, DC (IEEE), pp. 276–285.

Haniotakis, Th., Tsiatouhas, Y., and Nikolos, D., 1998, C-Testable one-dimensional ILAs with respect to path delay faults: theory and applications. *Proceedings of the IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems,* pp. 155–163.

Haniotakis, Th., Vergos, H. T., Tsiatouhas, Y., Nikolos, D., and Nicokaidis, M., 1999, Easily testable carry-save multipliers with respect to path delay faults. *Proceedings of the 2nd Electronic Circuits and Systems Conference (ECS '99)*, Bratislava, Slovakia, pp. 13–16.

Kalligeros, E., Vergos, H. T., Nikolos, D., Tsiatouhas, Y., and Haniotakis, Th., 1999, Path delay fault testable modified booth multipliers. *Proceedings of the XIV Design of Circuits and Iintegrated Systems Conference (DCIS '99), Palma de Mallorca,* pp. 16–19.

Lam, W. K., Saldanha, A., Brayton, R. K., and Sangiovanni-Vincitelli, A. L., 1993, Delay fault coverage and performance trade-offs. *Proceedings of the 30th Design Automation Conference,* Dallas, TX (ACM/IEEE Computer Society Press), pp. 446–452.

Lesser, J. D., and Shedletsky, J. J., 1980, An experimental delay test generator for LSI logic. *IEEE Transactions on Computers*, **C–29**, pp. 235–248.

Li, W. N., Reddy, S. M., and Sahni, S. K., 1989, On path selection on combinational logic circuits. *IEEE Transactions on Computer–Aided-Design,* pp. 56–63.

Lin, C. J., and Reddy, M., 1987, On delay fault testing in logic circuits. *IEEE Transactions on Computer-Aided Design,* pp. 694–703.

Nikolos, D., Haniotakis, Th., Vergos, H. T., and Tsiatouhas, G., 1999, Path delay fault testing of ICs with embedded intellectual property blocks. *Proceedings of Design, Automation and Test in Europe Conference (DATE '99),* Munich, Germany (IEEE Computer Society Press), pp. 112–116.

Pomeranz, I., and Reddy, S. M., 1994, An efficient non-enumerative method to estimate the path delay fault coverage in combinational circuits. *IEEE Transactions on CAD/ICAS,* **13**, 240–250.

Pomeranz, I., and Reddy, S. M., 2000, On *n*-detection test sets and variable *n*-detection test sets for transition faults. *IEEE Transactions on CAD/ICAS,* **19**, 372–383.

Pramanick, A. K., and Reddy, S. M., 1990, On the design of path delay fault testable combinational circuits. *Proceedings of the IEEE Fault Tolerant Computing Symposium,* pp. 374–381.

Sidiropoulos, G., Vergos, H. T., and Nikolos, D., 1999, Easily path delay fault testable non-restoring cellular array dividers. *Proceedings of the 8th Asian Test Symposium (ATS '99),* pp. 47–52.

Smith, G. L., 1985, Model for delay faults based upon paths. *Proceedings of the International Test Conference,* Philadelphia, PA (IEEE Computer Society Press), pp. 342–349.

Sparmann, U., Luxenburger, D., Cheng, K. T., and Reddy, S. M., 1995, Fast identification of robust dependent path delay faults. *Proceedings of the 32nd Design Automation Conference,* San Francisco, CA (IEEE/ACM), pp. 119–125.

Tani, S., Teramoto, M., Fukazawa, T., and Matsuhiro, K., 1999, Efficient path selection for delay testing based on partial path evaluation, *Journal of Electronic Testing: Theory and Applications,* **15**, pp. 75–85.

Vergos, H. T., Bellos, M., and Nikolos, D., 1999, Path delay fault testing of benes multistage interconnection networks. *Proceedings of the 6th IEEE International Conference on Electronics, Circuits and Systems (ICECS '99),* Pafos, Cyprus, Vol. II, pp. 1097–1100.

Weste, N. H. E., and Eshraghian, K., 1993, *Principles of CMOS VLSI Design* (Addison-Wesley), 2nd edition.

Williams, T. W., Underwood, B., and Mercer, M. R., 1991, The interdependence between delay optimization of synthesised networks and testing. *Proceedings of the 28th Design Automation Conference,* San Francisco, CA (ACM/IEEE), pp. 87–92.