Published at IEEE European Test Workshop (Poster)
Sete, France, June 12-14, 1996

Local Transformations and Robust Dependent Path Delay Faults **

(Extented Abstract)

Harry Hengster* Uwe Sparmann† Bernd Becker* Sudhakar M. Reddy‡

* Institute of Computer Science, Albert-Ludwigs-University, Freiburg, Germany
† Computer Science Department, University of Saarland, Saarbrücken, Germany
‡ Department of ECE, University of Iowa, Iowa City, USA

Abstract

Local transformations are used in several synthesis approaches. During application of such transformations attention has to be paid to many important properties, e.g. area, speed, power consumption, and testability. In this paper we study relations between local transformations and delay fault testability. In delay testing it is not necessary to test every path in a circuit to ascertain correct timing behavior. For example, a set of robust dependent path delay faults need not to be considered for testing if all paths that are not robust dependent are tested. We present sufficient conditions for local transformations which ensure that a test set for all non-robust-dependent paths in the original circuit is also a test set for all non-robust-dependent paths in the transformed circuit. These conditions are applied to some local transformations which are often used in logic synthesis and it is shown that they preserve testability. The impact of local transformations on robust dependent testability is demonstrated by experimental results performed on benchmark circuits.

1 Introduction

A *local transformation* is the substitution of a subcircuit by a different realization. Such transformations are often used in logic synthesis to optimize a given design with respect to the current goal of optimization. Besides the optimization of area, delay, and power consumption one of the most important aspects during logic synthesis is testability.

Thus, testability properties of local transformations have also been examined considerably in the literature. In [1] a synthesis approach for area optimization was presented and it was shown that the proposed transformations preserve stuck-at fault testability, i.e. if the original circuit is testable then the transformed circuit is testable.

**This work has been supported by DFG grants Be 1176/8-2 (H. Hengster) and SFB 124 – "VLSI Entwurfsmethoden und Parallelität" (H. Hengster and U. Sparmann) and by von Humboldt Foundations Senior Research Fellowship (S.M. Reddy).

For high speed circuitry not only logical faults but also delay faults have to be considered to check the correct behavior of the circuit. This led to the definition of the path delay fault model [2]. Initially, it was assumed that all possible path delay faults in a circuit have to be checked by *robust tests* [3] to ascertain correct timing behavior of the circuit. In this paper we use the term *complete path* (CP) testability, if testability is measured by the ratio of robustly testable paths to *all* paths in the circuit. It has already been observed in [4] that CP testability is quite poor for typical combinational circuits. In [5, 6] it has been shown that all but one of the local transformations of [1] also preserve CP testability, i.e. if all paths in the original circuit are robustly testable then all paths in the transformed circuit are robustly testable. Some of these transformations even improve CP testability [5]. In [7] SALT (System for Application of Local Transformations) was used to improve CP testability by local transformations. In [8] similar local transformations were suggested to improve testability.

Although, CP testability can be further improved considerably by local transformations it is still very low for most circuits. One source for low testability is the assumption that all possible path delay faults have to be considered. But it has been shown in [9] and [10] that the set of *robust dependent* (RD) path delay faults need not to be considered in order to ascertain correct timing behavior of a circuit. Thus, it is only necessary to robustly test the non-RD paths. We use the term $\overline{\text{RD}}$ testability, if testability is measured by the ratio of robustly testable paths to *non-RD* paths. Experimental results have shown that $\overline{\text{RD}}$ testability results in higher fault coverages than CP testability. A heuristic to determine RD paths which is applicable to large circuits was presented in [11].

In our paper we examine the effect of local transformations upon $\overline{\text{RD}}$ testability. We present sufficient conditions which ensure that a local transformation preserves $\overline{\text{RD}}$ testability, i.e. if all non-RD paths in the original circuit are robustly testable then all non-RD paths in the transformed circuit are robustly testable.

These conditions are more complex than the conditions for CP testability, because the supposition that all path delay faults of the original circuit are testable does not hold for $\overline{\mathrm{RD}}$ testability. Nevertheless, the presented conditions also refer only to the subcircuits describing a local transformation. Once the conditions are verified for a local transformation this transformation can be performed at any location in any circuit without deteriorating $\overline{\mathrm{RD}}$ testability of the circuit. Furthermore, the transformed circuit is tested by the same test set as the original circuit. Thus, there is no need to construct new test sets by pattern generation after local transformations have been applied.

After the conditions for $\overline{\mathrm{RD}}$ testability are presented, we show by application of these conditions that all but one of the local transformations of [1] preserve $\overline{\mathrm{RD}}$ testability.

Most practically used circuits are neither fully testable with respect to CP testability nor with respect to $\overline{\mathrm{RD}}$ testability. It has already been observed in [7] that local transformations improve CP testability. We give experimental results which demonstrate that application of local transformations to benchmark circuits does also improve $\overline{\mathrm{RD}}$ testability.

2 Basic Definitions

For simplicity of presentation we will restrict to single output combinational circuits in the following. For multi output circuits the theory is applied for each output cone separately.

The following definitions concerning delay testing are taken from [11]. A combinational circuit C consists of leads and gates. As gate types we will consider simple gates (AND, OR, NAND, NOR, and NOT), as well as primary inputs (PIs) and primary outputs (POs). A lead is connecting two gate pins with each other.

A *physical path* $p = (g_0, l_0, g_1, \ldots, l_{m-1}, g_m)$ in C is an alternating sequence of gates and leads leading from a PI g_0 to a PO g_m of C. We will associate two *logical paths* with each physical path. A logical path is given as a tuple (t, p) with $t = \overline{x} \rightarrow x,\ x \in \mathbf{B} = \{0, 1\}$, being a transition at the PI of p.

A manufactured implementation of circuit C is denoted by C_m. We assume that C_m has the gate level structure of C, but the delays of its gates and lines may take arbitrary values due to variations of the fabrication process or manufacturing defects. Let τ be the required operation time (clock period) for C_m. C_m is said to have a *delay fault* $> \tau$ on logical path (t, p) iff its delay for propagating t over p exceeds τ. A *robust test* for (t, p) [3] is a two pattern sequence which can be used to measure the delay of (t, p) in any implementation C_m of C. From the fact that C_m does (does not) operate correctly for a robust test under clock period τ it can be concluded that the delay of C_m for propagating t over p is $\leq \tau$ $(> \tau)$.

In [9, 10] it has been shown that in order to guarantee that the delay of a manufactured circuit meets its specification not all logical paths need to be checked robustly.

Definition 1 ([9]) *Let $\mathcal{LP}(C)$ be the set of all logical paths in a circuit C and $\mathcal{R}$ a subset of $\mathcal{LP}(C)$.*

$\mathcal{R}$ is said to be a robust dependent set (RD-set) *iff for all implementations C_m of C and all clock periods τ the following holds: The absence of delay faults $> \tau$ on the logical paths from $\mathcal{LP}(C) \setminus \mathcal{R}$ implies the delay of C_m is $\leq \tau$.*

From Definition 1 and the above stated property of robust tests it follows directly: For verifying that the delay of C_m does not exceed the clock period it is sufficient to check all non-RD paths $(\mathcal{LP}(C) \setminus \mathcal{R})$ with robust tests. A set of two pattern sequences, which contains a robust test for every logical path in $\mathcal{LP}(C) \setminus \mathcal{R}$ for some RD-set $\mathcal{R}$, is called a *full $\overline{RD}$ test set.*

The following definitions are important for the understanding of the following theorems (cf. Example 1).

Definition 2

i) For circuit C and input vector $v \in \mathbf{B}^n$ a stabilizing system *S is a minimal subcircuit of C such that S stabilizes the* PO *of C on its final (stable) value $C(v)$ independent of the circuitry of C not included in S.*

ii) The set of logical paths included in stabilizing system S for input vector v is denoted by $\mathcal{LP}(v, S)$.

iii) The set of all possible stabilizing systems for circuit C and for input vector v is denoted by $\mathcal{S}(v, C)$.

iv) A complete stabilizing assignment *σ is a mapping which assigns to each input vector $v \in \mathbf{B}^n$ a stabilizing system, denoted by $\sigma(v)$.*

v) The set of all logical paths included in the stabilizing systems of a complete stabilizing assignment σ is denoted by $\mathcal{LP}(\sigma)$.

There can exist many different stabilizing systems for one input vector v. An algorithm to compute the stabilizing systems was presented in [11]. Note, that a stabilizing system normally only constitutes a small portion of the overall circuit.

Example 1 *A circuit C realizing functions* `f` *and* `g` *is given in Figure 1. We will consider only the output cone of* `f` *for input vector $v = 000$. The stabilizing systems for this input vector are shown by bold lines. Each path drawn with bold lines represents the logical path with a falling transition. For input vector $v = \langle 000 \rangle$ there exist exactly three stabilizing systems S_1, S_2, and S_3 which form the set $\mathcal{S}(v, C)$ of all possible stabilizing systems.*

To obtain the final (stable) value 0 *at* PO `f` *of circuit C for input vector $v = \langle 000 \rangle$ it suffices to switch the paths included in one stabilizing system. For example in stabilizing system S_1 it suffices that a falling transition at* PI `a` *travels along path* `af` *in order to guarantee that the* PO `f` *obtains the final (stable) value* 0.

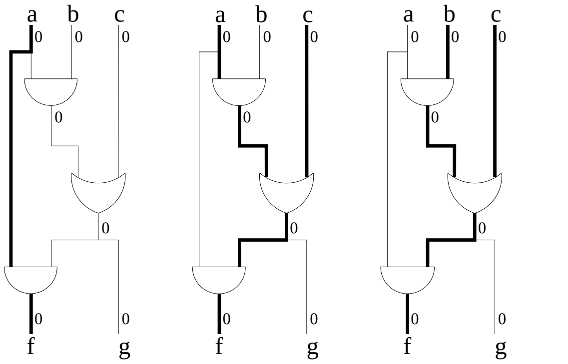

Figure 1: Stabilizing systems S_1, S_2, S_3 for $v = \langle 000 \rangle$

Now, we can present the theorem from [11] which shows that it is sufficient to only test the logical paths from $\mathcal{LP}(\sigma)$ robustly.

Theorem 1 ([11]) *Let $\mathcal{LP}(C)$ denote the set of all logical paths in C and consider an arbitrary complete stabilizing assigment σ for C. Then $\mathcal{LP}(C) \setminus \mathcal{LP}(\sigma)$ is an RD-set.*

The formal proof of the theorem can be found in [11]. It is based on the fact that it suffices to check the logical paths of one stabilizing system for each input vector of the circuit. The selection of the stabilizing systems is given by the complete stabilizing assignment σ.

Example 2 *Consider again the circuit C of Figure 1. The falling transition at the path starting at input* `a` *and running through all three gates is not robustly testable. But it is possible to choose a complete stabilizing assignment such that this logical path is not included in the stabilizing system for any $v \in \mathbf{B}^3$. Since all other logical paths of the circuit C are robustly testable, we conclude that C is fully $\overline{RD}$ testable.*

This example shows that it is crucial to consider $\overline{\mathrm{RD}}$ testability instead of CP testability, if the objective of a syntesis tool is to obtain fully delay testable and area optimal circuits. The circuit of Figure 1 is fully $\overline{\mathrm{RD}}$ testable. But, it can be shown that any realization, where all path delay faults are robustly testable, needs more gates to implement `f` and `g` simultaneously.

3 Local Transformations

A *local transformation* is the substitution of a subcircuit by a new realization which has better properties with respect to the current goal of optimization. Local transformations are often used in logic synthesis to improve the area and the depth of a given design while testability properties are retained. In [1] an algorithm for multi-level logic synthesis based on local transformations has been presented. In this paper it was also shown that these transformations preserve stuck-at fault testability. In [5, 6] it has been shown that all but one of these local transformations preserve CP testability. Some of these transformations even improve CP testability [5]. In [7] SALT (System for Application of Local Transformations) was used to improve CP testability by local transformations.

In the following theorem we present sufficient conditions, which ensure that a local transformation preserves $\overline{\mathrm{RD}}$ testability, i.e. if all non-RD paths in the original circuit are robustly testable then all non-RD paths in the transformed circuit are robustly testable.

A path mapping for the circuits describing the local transformation is used to derive stabilizing systems for the transformed circuit from the stabilizing systems of the original circuit. These stabilizing systems are used to argue about $\overline{\mathrm{RD}}$ testability of the transformed circuit.

Theorem 2 *Let T be a full $\overline{RD}$ test set for N_1 and let C_1 be a subcircuit of N_1 with m inputs and 1 output. T is a full $\overline{RD}$ test set for the circuit N_2, which is obtained from N_1 by substituting C_2 for C_1, if conditions* (i) - (iii) *hold for circuit C_2.*

(i) *C_2 has the same input sequence and realizes the same function as C_1.*

(ii) *There exists a mapping $\pi : \mathcal{LP}(C_1) \longrightarrow \mathcal{LP}(C_2)$:*
$$\forall v \in \mathbf{B}^m : \ \forall S_1 \in \mathcal{S}(v, C_1) : \ \exists S_2 \in \mathcal{S}(v, C_2) :$$
$$\mathcal{LP}(v, S_2) \subseteq \{\pi(p) \mid p \in \mathcal{LP}(v, S_1)\} \textit{ and}$$
$$\forall p \in \mathcal{LP}(v, S_1) : \textit{ Any robust test for } p \textit{ is a robust test for } \pi(p).$$

(iii) *If a two pattern sequence t produces a stable value at the output of C_1 then t produces a stable value at the output of C_2.*

Intuitively, condition (ii) allows to choose for each stabilizing system S_1 to circuit C_1 a stabilizing system S_2 to C_2 such that the following holds: Each full delay test for the logical paths of S_1 is also a full delay test for the logical paths of S_2.

The formal proof of the theorem is given in [12]. Since the conditions of Theorem 2 refer only to the substituted region of the circuit it suffices to examine only the representation of the subcircuits. Once the conditions are verified for a given local transformation this transformation can be performed at any location in any circuit without deteriorating $\overline{\mathrm{RD}}$ testability of the circuit. Additionally, the transformed circuit is tested by the same test set as the original circuit. Thus, there is no need to construct new test sets by pattern generation after local transformations have been applied.

Table 1 shows some local transformations which are wide-spread in logic synthesis. The circuit realizations can be found in [12]. The first and second local transformations describe factorization and De Morgan's law, respectively. The remaining five transformations correspond to different kinds of cube extraction as introduced in [1]. Note, that transformation $\mathcal{T}_2$ is a special case of factorization.

$$\mathcal{T}_F : A_1B_1 + \ldots + A_1B_m + \ldots + A_nB_1 + \ldots + A_nB_m \Rightarrow (A_1 + \ldots + A_n)(B_1 + \ldots + B_m)$$

$$\mathcal{T}_M : a_1 \cdot a_2 \cdot \ldots \cdot a_n \Rightarrow \overline{\overline{a_1} + \overline{a_2} + \ldots + \overline{a_n}}$$

$$\mathcal{T}_1 : (ab)A_1 + (ab)A_2 + \ldots + (ab)A_n \Rightarrow CA_1 + CA_2 + \ldots + CA_n \quad \text{with}^1 \quad C = ab$$

$$\mathcal{T}_2 : A_1B_1 + A_1B_2 \Rightarrow A_1(B_1 + B_2)$$

$$\mathcal{T}_3 : abA + \bar{a}B + \bar{b}B \Rightarrow MA + \overline{M}B \quad \text{with} \quad M = ab$$

$$\mathcal{T}_4 : (a\bar{b} + \bar{a}b)A + (\bar{a}\bar{b} + ab)B \Rightarrow MA + \overline{M}B \quad \text{with} \quad M = a\bar{b} + \bar{a}b$$

$$\mathcal{T}_5 : \bar{a}bA + acA + \bar{a}\bar{b}B + a\bar{c}B \Rightarrow MA + \overline{M}B \quad \text{with} \quad M = \bar{a}b + ac$$

Table 1: Local transformations

The conditions of Theorem 2 have been applied to these local transformations. Their effect upon $\overline{\text{RD}}$ testability is described by the following lemma:

Lemma 1

a) *The application of transformations $\mathcal{T}_F$, $\mathcal{T}_M$, and $\mathcal{T}_1$ – $\mathcal{T}_4$ to a subcircuit in a fully $\overline{RD}$ testable circuit preserves $\overline{RD}$ testability. Any full test set for the original circuit is also a full test set for the transformed circuit.*

b) *The application of transformation $\mathcal{T}_5$ to a subcircuit in a fully $\overline{RD}$ testable circuit does not preserve $\overline{RD}$ testability in general.*

The proof of the lemma can be found in [12]. To prove part b) a circuit is presented, which initially is fully $\overline{\text{RD}}$ testable. After an instance of $\mathcal{T}_5$ is applied to this circuit the circuit is no longer fully $\overline{\text{RD}}$ testable .

Although $\mathcal{T}_5$ does not preserve $\overline{\text{RD}}$ testability in general, the lemma does not prove that testability of a circuit is always destroyed by application of $\mathcal{T}_5$. For many instances of the transformation testability will still be preserved. It is also interesting to note that $\mathcal{T}_5$ was the only transformation which did not preserve CP testability [5, 6].

Furthermore, it should be mentioned that in some cases only a subset of the original $\overline{\text{RD}}$ test set is needed to test the transformed circuit. It is also possible that the transformed circuit is fully $\overline{\text{RD}}$ testable even if the original circuit was not. In this sense the local transformations do even improve testability. In the following section we will show that local transformations considerably improve $\overline{\text{RD}}$ testability for typical combinational circuits.

[1]The expression $C = ab$ is evaluated n times for the original circuit and only once for the transformed circuit.

In addition to the local transformations already presented the absorbtion rule $a \cdot (a + b) = a$ and Boolean reductions like $a \cdot a = a$, $a \cdot \overline{a} = 0$, etc. can improve $\overline{\text{RD}}$ testability and the size of the transformed circuits. Moreover, for every local transformation $\overline{\text{RD}}$ testability properties can be directly transfered to the dual transformation [13].

4 Experimental Results

In this section we examine the effect of local transformations on $\overline{\text{RD}}$ testability for some ISCAS85 circuits [14] and the combinational parts of some ISCAS89 circuits [15].

In Table 2 we give two rows for each circuit. In the first row we give the statistics for the original benchmark circuit. The second row shows the statistics for the circuits which are optimized by SALT [7]. We applied the same local transformations which were used in [7] to improve CP testability. The results can be improved if the selection among instances of local transformations which exclude one another is directed by special strategies for $\overline{\text{RD}}$ testability.

In the first column of Table 2 the name of the circuit is given. The number of logical paths in the circuit is put in the second column. The third number gives the number of logical non-RD paths determined by the heuristic of [11]. These paths have to be tested by robust tests to verify that the delay of the circuit does not exceed the clock period. The number of logical path, which are testable by robust tests, is given in the fourth column.

The next two columns give the fault coverages with respect to CP testability (CP-FC) and $\overline{\text{RD}}$ testability ($\overline{\text{RD}}$-FC), respectively. CP fault coverage is computed by dividing the number of testable paths by the number of all logical paths in the circuit. To calculate $\overline{\text{RD}}$ fault coverage only the non-RD paths are considered instead of the number of all logical paths.

The advantages of local transformations can be seen in Table 2. For CP testability as well as for $\overline{\text{RD}}$ testability the fault coverage is increased by local transformations, i.e. more untestable, non-RD paths than robustly testable paths are removed. For example for circuit cs38584 CP fault coverage is increased from 4.27% to 18.36% and $\overline{\text{RD}}$ fault coverage is improved by local transformations from 20.61% to 41.78%. Also for circuits with high initial testability (c0880) local transformations can still improve testability considerably.

In the last two columns the number of gates and the depth is given for each circuit. These numbers are determined for circuit realizations which consist of 2-input gates only. Application of local transformations does not only improve testability but also the depth remains nearly the same and the number of gates is even decreased.

circuit		log. paths	non-RD	testable	CP-FC	$\overline{\text{RD}}$-FC	gates	depth
c0880	orig.	17284	16731	16083	93.05	96.13	346	25
	transf.	15092	14834	14438	95.67	97.33	340	27
c5315	orig.	2682610	433596	81435	3.04	18.78	2079	37
	transf.	2081924	347042	70000	3.36	20.17	1876	37
cs00713	orig.	43624	4996	1184	2.71	23.70	198	37
	transf.	35134	4722	1126	3.20	23.85	182	36
cs13207	orig.	2690738	527998	27603	1.02	5.23	3214	36
	transf.	855740	325120	35432	4.14	10.90	3023	33
cs38584	orig.	2161446	447496	92239	4.27	20.61	13502	37
	transf.	571982	251428	105038	18.36	41.78	13022	37

Table 2: Effect of local transformations upon testability

5 Conclusions

Sufficient conditions which ensure that a local transformation preserves $\overline{\text{RD}}$ testability have been presented. The conditions refer only to the subcircuits which describe the local transformation. Once the conditions are checked for a local transformation this transformation can be performed at any location in any circuit. A test set for the original circuit will also test the transformed circuit.

Using these conditions we proved for a set of local transformations which are frequently used in logic synthesis that they preserve $\overline{\text{RD}}$ testability. Experimental results showed that these transformations also improve $\overline{\text{RD}}$ testability for circuits which are not fully $\overline{\text{RD}}$ testable.

References

[1] J. Rajski and J. Vasudevamurthy. The testability - preserving concurrent decomposition and factorization of boolean expressions. *IEEE Trans. on CAD*, 11(6):778–793, 1992.

[2] G.L. Smith. Model for delay faults based upon paths. In *Int'l Test Conf.*, pages 342–349, 1985.

[3] C.-J. Lin and S.M. Reddy. On delay fault testing in logic circuits. *IEEE Trans. on CAD*, 6(5):694–703, 1987.

[4] S.M. Reddy, C.J. Lin, and S. Patil. An automatic test pattern generator for the detection of path delay faults. In *Int'l Conf. on CAD*, pages 284–287, 1987.

[5] H. Hengster, R. Drechsler, and B. Becker. Testability properties of local circuit transformations with respect to the robust path-delay-fault model. In *VLSI Design Conf.*, pages 123–126, 1994.

[6] A. El-Maleh and J. Rajski. Delay-fault testability preservation of the concurrent decomposition and factorization transformations. In *VLSI Test Symp.*, pages 15–21, 1994.

[7] H. Hengster, R. Drechsler, and B. Becker. On local transformations and path delay fault testability. *Jour. of Electronic Testing: Theory and Applications*, 7(3):173–191, 1995.

[8] I. Pomeranz and S.M. Reddy. On the number of tests to detect all path delay faults in combinational logic circuits. *IEEE Trans. on Comp.*, 45(1):50–62, 1996.

[9] W.K. Lam, A. Saldanha, R.K. Brayton, and A.L. Sangiovanni-Vincentelli. Delay fault coverage and performance tradeoffs. In *Design Automation Conf.*, pages 446–451, 1993.

[10] K.-T. Cheng and H.-C. Chen. Delay testing for nonrobust untestable circuits. In *Int'l Test Conf.*, pages 954–961, 1993.

[11] U. Sparmann, D. Luxenburger, K.-T. Cheng, and S.M. Reddy. Fast identification of robust dependent path delay faults. In *Design Automation Conf.*, pages 119–125, 1995.

[12] H. Hengster, U. Sparmann, B. Becker, and S.M. Reddy. Local transformations and robust dependent path delay faults. Technical report, Albert-Ludwigs-University, Freiburg, 1996.

[13] S. Kundu and A.K. Pramanick. Testability preserving boolean transforms for logic synthesis. In *VLSI Test Symp.*, pages 131–138, 1993.

[14] F. Brglez and H. Fujiwara. A neutral netlist of 10 combinational circuits and a target translator in fortran. In *Int'l Symp. Circ. and Systems, Special Sess. on ATPG and Fault Simulation*, pages 663–698, 1985.

[15] F. Brglez, D. Bryan, and K. Kozminski. Combinational profiles of sequential benchmark circuits. In *Int'l Symp. Circ. and Systems*, pages 1929–1934, 1989.