ROBDD Based Path Delay Fault Testable Combinational Circuit Synthesis

Toral Shah, Virendra Singh
CADSL, Dept. of Electrical Engineering
Indian Institute of Technology Bombay
{toral77, viren}@ee.iitb.ac.in

Anzhela Matrosova
Dept. of Applied Mathematics and Cybernetics
Tomsk State University
mau11@yandex.ru

Abstract

Traditional scan based transition delay fault tests can potentially miss variability induced delay faults on long interconnects. On the other hand, an ATPG may not be successful in deriving test patterns for all paths. The paper proposes a BDD based synthesis method where all the paths are testable under the path delay fault model without addition of extra inputs. Each ROBDD (Reduced-Ordered-Binary Decision Diagram) node is covered by an Invert-AND-OR sub-circuit. The paper proves that the synthesized circuit is fully testable for path delay faults, either by robust tests or validatable non-robust tests.

1. Introduction

Integrated circuits implemented in deep sub-micron technology usually suffer from subtle timing violations due to process variations on both gates and interconnects. These variations have minimal impact on path delays of short interconnects but significant impact on path delays of very long interconnects whose resistance can be as high as a *kohm* or even greater.

To observe delay defects, it is necessary to generate transitions at the circuit input and propagate them to the outputs. This requires application of a pair of vectors (v_1, v_2). The first vector v_1 stabilizes all signals in a circuit. The second vector v_2 causes the desired transition at the input of the circuit. Transition delay fault model detects only gross delays. On the other hand *Path Delay Fault* (PDF) model takes into account distributed delays over the paths, hence it is a better suited model for delay test. In accordance with the conditions of fault manifestation, singly testable PDFs are divided into robust testable faults and non-robust testable faults [1].

The number of paths in a circuit are in general very large. This may lead to the ATPG not being able to generate tests or gets timed out. To overcome this issue, addition of multiplexers [2] or gates [3] on the dependent off paths have been proposed. The disadvantages of these approaches are:

- It takes long time for the ATPG to declare that no test exists for a particular path.
- This is a post design-phase procedure, hence any circuit changes will have to be followed by timing analysis.
- Additional control inputs are needed [2]-[5]. This would result in additional input pins.

In such scenarios, there is a pressing need to develop synthesis techniques which keep testability on the forefront. If the circuit is designed in such a way that the delay testability of all paths is guaranteed then the disadvantages discussed above can be overcome.

In [4], a simple transformation of a circuit is suggested that guarantees 100% testability for both single stuck-at fault (SAF) and PDF models. These circuits are derived from ROBDDs using muxes. The size of a circuit is proportional to the given ROBDD node count. The major disadvantage of the approach in [4] is the use of additional control input which leads to an increase in the number of the chip pins.

EX-OR Projected Sum-of-Products (EXSOPs)[5] is another synthesis method where the paths are delay testable. Symmetric circuits synthesized by a method that guarantees complete path delay testability without any additional control inputs, was proposed in [6].

In paper [9] the circuits constructed from ROBDDs by special sub-circuit implementation using Shannon expansion:

$$f_v = \bar{x_i} \cdot f_v^{x_i=0} \oplus x_i \cdot f_v^{x_i=1} \tag{1}$$

represent an internal node v of the ROBDD. In this formula the operation $\oplus$ is realized by XOR gate. It is proved that *each path delay fault of the resulted circuit manifests itself as robust testable fault* [9]. When applying the test pairs in the definite order we can detect any PDF of the circuit. Thus this synthesis technique guarantees 100% testability of the corresponding circuits for PDFs without an additional input. However, the path length increases considerably due to presence of XOR gates.

978-1-5090-0693-9/16/$31.00 ©2016 IEEE

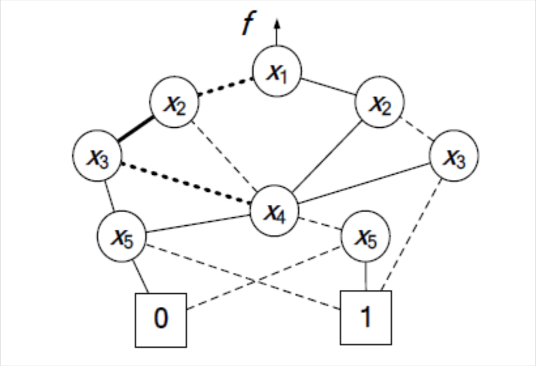

Fig. 1. A ROBDD example for a Boolean function f

In this proposal the circuit is constructed from ROBDD by covering each internal node v with sub-circuit implementing the simpler Shannon expansion:

$$f_v = \bar{x_i} \cdot f_v^{x_i=0} + x_i \cdot f_v^{x_i=1} \tag{2}$$

In this formula the operation $+$ is realized by OR gate. As a result of this implementation, the path delay faults manifest themselves either as robust testable or validatable non-robust testable ones. It means that this synthesis approach too guarantees 100% testability for essential PDFs without an additional input.

The main contribution of this paper is that it proposes a fully delay testable circuit without any additional input. An algorithm to derive the test vector pairs for detecting the path delay faults is presented.

Rest of the paper is organized as follows. The procedure of circuit design is given in Section 2. Section 3 describes the test pattern generation procedure for stuck-at faults and path delay faults. Experimental results are presented in Section 4. Section 5 concludes the paper.

2. Combinational Circuit Design

Reduced and ordered BDD (ROBDD) is a canonical representation of Boolean function for the chosen order of variables [8]. Without loss of generality we consider single output function in this work for simplicity and explanation. Figure 1 represents the ROBDD for one output Boolean function.

When all edges directed towards 0-terminal node are eliminated, resultant ROBDD depicts combinational circuit behavior. Each node so derived from ROBDD is covered with the sub-circuit implementing the Shannon decomposition formula (Invert-And-Or sub-circuit) shown in Figure 2. For nodes with one edge connected to the terminal node, reduction using boolean algebra must be avoided. Simplification leads to untestable design at next level [4]. For instance, in Figure 1, the node x_3 on the right hand side has one edge connected to terminal node 1. Hence this has been implemented as $\bar{x_3} + x_3 f_{x_4}$ in Figure 3.

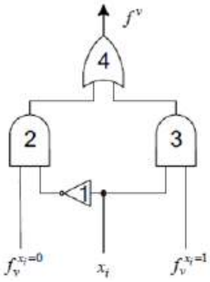

Fig. 2. Gate implementation of $f_v = \bar{x_i} \cdot f_v^{x_i=0} + x_i \cdot f_v^{x_i=1}$

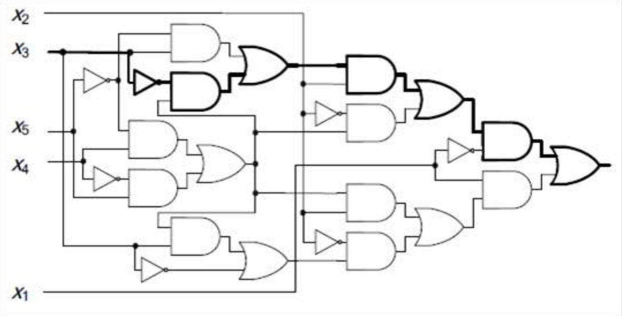

Fig. 3. Circuit C corresponding to function f in Fig. 1

There are four conditions where one of the node edges is connected to leaf node 0 or 1. If the edge of a node is connected to 0, then the sub-circuit can be simplified to an AND gate without disturbing the testability. If one of the node edges is connected to leaf node 1 then the sub-circuit simplifies to an OR gate. This will cause a transition on the control variable to be masked by a 1 on the other input of the OR gate.

$$\bar{x_i} + f_{x_j} = \bar{x_i} + x_i \cdot f_{x_j} \tag{3}$$

$$x_i + f_{x_j} = x_i + \bar{x_i} \cdot f_{x_j} \tag{4}$$

To avoid such situations, the implementation shown in equations above should be used. This transformation maintains the testability of paths running through the corresponding nodes by making sure that the OR gate input at the off path has a non-controlling value. As a result of covering the circuit ROBDD we generate the combinational circuit C shown in Figure 3.

3. Test Pattern Generation

A Boolean expression can be represented as a sum of disjoint cubes also called as DSoPs (disjoint sum-of-products). All the individual paths from root node to leaf node 1 are disjoint and constitute a DSoP [11]. The DSoP of function F (Figure 1) is as follows:

$$\begin{aligned} F = {} & x_1\bar{x_2}\bar{x_3} + x_1\bar{x_2}x_3\bar{x_4}x_5 + x_1\bar{x_2}x_3x_4\bar{x_5} \\ & + x_1x_2\bar{x_4}x_5 + x_1x_2x_4\bar{x_5} + \bar{x_1}\bar{x_2}\bar{x_4}x_5 \\ & + \bar{x_1}\bar{x_2}x_4\bar{x_5} + \bar{x_1}x_2\bar{x_3}\bar{x_4}x_5 + \bar{x_1}x_2\bar{x_3}x_4\bar{x_5} \\ & + \bar{x_1}x_2x_3\bar{x_5} \end{aligned} \tag{5}$$

The DSoP representation is crucial to the test pattern generation. The sequence of inputs encountered on the path to be tested is contained in at least one of the product terms.

3.1 Stuck-at Faults

Consider stuck-at faults at a primary input or at the gate which falls on a path of the circuit C (shown in Figure 3). Let this path be represented by product K from the DSoP and let x_i be the primary input included in the path. Let $K^{'}$ be a product term obtained by complementing x_i to $\bar{x_i}$. It is proved in [13] that each fault of the DSoP $x_i = 0$ is detectable but fault $x_i = 1$ will be detected only if $K^{'}$ is not an implicant of the DSoP.

3.2 Path Delay Faults

The conditions of robust path delay fault manifestation for test pairs have been formulated by Matrosova et.al. [10]. The conditions are as follows:

1) There exists a test pattern for the corresponding stuck-at fault of the literal. The test pattern is vector v_2 of the test pair (v_1, v_2).
2) The variable x_i that marks beginning of the path takes complementary values for v_1, v_2 vectors.

Consider the path α that begins at the input marked by a literal x_i, (for the sake of definiteness the input is marked with a literal x_i without inversion) and terminates at the circuit output. The s-a-1 fault at x_i corresponds to falling transition of the path α and s-a-0 fault at x_i corresponds to rising transition [9].

Let a path starting from the root node of the ROBDD to the node marked with x_i be called ε. Sub-product of the path ε is k_ε. We may find several prolongations and consequently several products containing the literal x_i corresponding to the beginning of the path α. All these products comprise a set K_α [10]. The paths controlled by x_i and $\bar{x_i}$ in the same sub-circuit, are called *companion* paths. Gates 1-2-4 and 3-4 in Figure 2 form companion paths. Note that the products of the DSoP that contain the literal $\bar{x_i}$ corresponding to the companion path of α have the same sub-product k_ε.

Theorem 1. *If k_γ is a sub-product on which the condition $f_v^{x_i=0} \neq f_v^{x_i=1}$ occurs, especially $f_v^{x_i=0} = 0$ and $f_v^{x_i=1} = 1$ and the product $k_\varepsilon k_\gamma$ (excluding $x_i, \bar{x_i}$) is represented by k^*, the product k^* represents a test pair which can robustly test path α for rising and falling transitions.*

Proof. The product $x_i k^*$ represents the Boolean vector v_2 that turns the product K from K_α to 1. It means v_2 is a test pattern for the fault. The product $\bar{x_i} k^*$ represents vector v_1 of this test pair. The product v_1 turns $K^{'}$ into 1 and turns DSoP into 0. The test pair detects rising transition of the path α. Actually v_1 is orthogonal to all products of the DSoP that do not contain sub-product k_ε and $\bar{x_i}$ as k_γ is orthogonal to $f_v^{x_i=0}$. To test for falling transition v_1 must be taken as v_2 and vice versa for the same path α. None of the DSoP products contain repeated literals. Thus all conditions of robust testable manifestation for rising and falling transition of the path α are fulfilled. Hence the theorem is proved. □

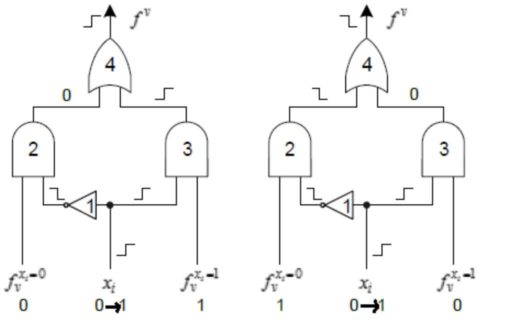

Fig. 4. Robustly testable conditions

Corollary 1. *If there exists a path γ for which $f_v^{x_i=1}(\gamma) = 1, f_v^{x_i=0}(\gamma) = 0$ and a path δ for which $f_v^{x_i=1}(\delta) = 0, f_v^{x_i=0}(\delta) = 1$ then for both paths that begin at the same input and marked with the literals x_i and $\bar{x_i}$, PDFs manifest themselves as robust testable for rising and falling transitions.*

Figure 4 illustrates the above mentioned corollary.

Example: Refer the bold path in Figure 3. ($k_\varepsilon = \bar{x_1} x_2 \bar{x_3}$). This sub path exists in the 8^{th} term and the 9^{th} term of the DSoP (Equation 5). Picking the 8^{th} term $\bar{x_1} x_2 \bar{x_3} \bar{x_4} x_5$ gives $v_2 = 01001$. To test the path via x_3, the value of x_3 is complemented to get $v_1 = \bar{x_1} x_2 x_3 \bar{x_4} x_5 = 01101$. v_1 is not an implicant of the DSoP. Thus $(v_1, v_2) = (01101, 01001)$ will test the path for rising transition via x_3. Interchanging v_1 & v_2 will test the path for falling transition via x_3.

For internal nodes both conditions, i.e., $f_v^{x_i=0}(\gamma) = 0, f_v^{x_i=1}(\gamma) = 1$ and $f_v^{x_i=0}(\delta) = 1, f_v^{x_i=1}(\delta) = 0$ exist. There are some nodes where only one of the condition is true. This happens when one of the functions $f_v^{x_i=0}, f_v^{x_i=1}$ is an implicant of the other, i.e., $f_v^{x_i=0} \leq f_v^{x_i=1}$ or $f_v^{x_i=1} \leq f_v^{x_i=0}$. It is possible that only one condition pointed in the corollary holds good. In that case, the path which is robustly testable should be tested first and then proceed to test the companion path. Let α be the companion path.

Theorem 2. *Let there be an ROBDD node, wherein the condition $f_v^{x_i=1} = 0, f_v^{x_i=0} = 1$ does not exist. Also let k_γ be a sub-product on which the condition $f_v^{x_i=0} = 1$ and $f_v^{x_i=1} = 1$ exists. Product $k_\varepsilon k_\gamma$ (excluding $x_i, \bar{x_i}$) is represented by k^*. The product k^* represents a test pair which can non-robustly test path α for rising transition.*

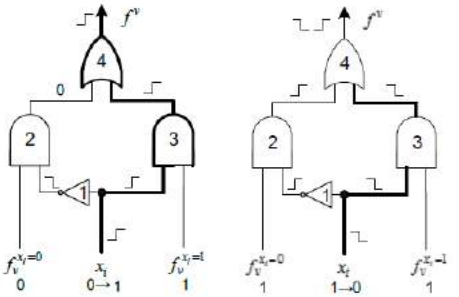

Fig. 5. Non-robustly testable conditions

Proof. The product $x_i k^*$ represents the Boolean vector v_2 that turns the product K from K_α to 1. It means v_2 is a test pattern for the fault $x_i = 0$. The product $\bar{x}_i k^*$ represents vector v_1 of this test pair. The test pair detects rising transition of the path α. The product v_1 turns k^* to 1. Moreover as the condition $f_v^{x_i=1}(\delta) = 0, f_v^{x_i=0}(\delta) = 1$ is not feasible, the product K^* is an implicant of DSoP and consequently v_1 is not a test pattern for the fault. Hence the theorem is proved. □

4. Experimental results

Table I provides gate count comparison between the Invert-AND-XOR and Invert-AND-OR based circuit synthesis. To derive the optimal ROBDD, we have used the *abc* tool [12] which has an underlying *CUDD* package.

First column of Table I lists used combinational benchmark circuits, the second column indicates their inputs and outputs. The third column lists the number of ROBDD nodes after optimal ordering. The fourth column shows the gate count (two-input gates) if the XOR based sub-circuit [9] is used to implement each ROBDD node and the fifth column indicates the gate count(two-input gates) if the OR sub-circuit discussed in this paper is used to cover ROBDD nodes. The XOR sub-circuit would comprise of 5 2-input gates whereas the OR sub-circuit is constructed using 3 2-input gates. Thus ROBDD based circuits implemented using OR based modules would have a 40% lesser gate count as compared to the circuits constructed using XOR based modules.

TABLE I
Gate count comparison

Benchmark circuit	Ips/Ops	ROBDD nodes	Gate count as per [9]	Gate count as per current proposal
alu4	14/18	5372	26860	16116
apex2	39/3	6713	33565	20139
C3540	50/22	1747	8735	5241
C5315	178/123	2387	11935	7161
C6288	32/32	2389	11945	7167
C7552	207/108	2328	11640	6984
dalu	75/16	1989	9945	5967
i10	257/224	3032	15160	9096
misex3	14/14	4995	24975	14985
pdc	16/40	16621	83105	49863
rot	135/107	1015	5075	3045

For each node, the path length from x_i to the sub-circuit output would be 3 gates for XOR based and 2 gates for an OR based sub-circuits respectively. Hence the overall path length (of all paths including the longest path) would reduce by 33% if OR based sub-circuits are used over XOR based sub-circuits.

5. Conclusion

This paper proposed an ROBDD based combinational circuit synthesis methodology. Each ROBDD node is implemented using an Invert-And-Or sub-circuit. *This method eliminated the need of additional control inputs, required for easier testability, in prior methods.* Also path lengths as compared to XOR based implementation, are reduced. All paths are proved to be testable for *path delay faults* either robustly or by validatable non-robust tests.

References

[1] M.L. Bushnell, and V.D. Agrawal, ”Essentials of electronic testing for digital, memory and mixed-signal VLSI Circuits”, *Kluwer Academic Publishers*, Boston, Mass, USA, 2000.

[2] I. Pomeranz, and S. Reddy, “Design-for-testability for improved path delay fault coverage of critical paths”, *21st International Conference VLSI*, pp.175-180, 2008.

[3] M. Siebert, and E. Gramatova. “Delay Fault Coverage Increasing in Digital Circuits”, *Euromicro Conference on Digital System Design (DSD)*, pp. 475-478, 2013.

[4] R. Dreshler, J. Shi and G. Fey, “Synthesis of Fully Testable Circuits from BDDs”, *IEEE Transactionss on Computer-Aided Design of Integrated Circuits and Systems*, Vol.23, No. 3, pp.1-4, 2004.

[5] A. Bernasconi, V. Ciriani, R. Cordone, “EXOR Projected Sum of Products”, *IFIP International Conference on Very Large Scale Integration VLSI-SOC*, pp.284-289, 2006

[6] S. Chakrabarti, S. Das; D. K. Das,B. B. Bhattacharya, “Synthesis of symmetric functions for path-delay fault testability”, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* Vol. 19, No.9, pp.1076-1081, 2000

[7] B. Becker, “Testing with decision diagrams”, *Integration, the VLSI Journal*. vol. 26, pp. 5-20, 1998

[8] R. E. Bryant, “Graph-based algorithms for Boolean function manipulation”, *IEEE Trans. on Computers*, pp. 677-691, 1986

[9] A. Matrosova, and E. Nikolaeva, “PDFs testing of combinational circuits based on covering ROBDDs”, *East West Design and Test Symposium*, pp. 160-163, 2010.

[10] A. Matrosova, V. Lipsky, A. Melnikov, and V. Singh, “Path delay faults and ENF”, *EWDTS*, pp.164-167, 2010.

[11] N. Dreschler, M. Hilgemier, G. Fey and R. Dreschler, “Disjoint Sum of Product Minimization by Evolutionary Algorithms”, Chapter *Applications of Evolutionary Computing*, Lecture Notes in Computer Science, Vol. 3005, pp.198-207, 2004

[12] Berkeley Logic Synthesis and Verification Group, ABC: A System for Sequential Synthesis and Verification, Release 20160618. http://www.eecs.berkeley.edu/ alanmi/abc/

[13] T. Shah, A. Matrosova, V. Singh,” BDD based PDF Testable Combinational Circuit Design”, WRTLT , India, 2015