ISSN:2229-6093

Algorithm for Determining Most Qualified Nodes for Improvement in Testability

Rupali Aher, Sejal Badgujar, Swarada Deodhar and P.V. Sriniwas Shastry,

Department of Electronics and Telecommunication, MKSSS's Cummins College of Engineering for Women
Pune, India.
Email: rupali.aher@cumminscollege.in pvs.shastry@cumminscollege.in

Abstract— **This paper proposes an algorithm for testability measurement of combinational circuits and determination of Most Qualified Nodes for Improvement in testability (MQNI). The algorithm is verified on the benchmark combinational circuits of ISCAS'89. The netlist (Verilog) file is directly used as an input to this algorithm and a list of nodes for insertion of effective Design for Testability (DFT) is generated. A table of Testability Improvement Factor (TIF) after insertion of DFT is also generated, at the shortlisted nodes without realization of actual schematic circuit. The algorithm is based on SCOAP [1], to measure the testability of the circuit. Testability improvement of every node is considered individually and analyzed on the basis of TIF hence computed. This algorithm makes use of testability and fan out of a node to identify MQNI. Improvement in testability and area overhead for various benchmark circuits are obtained by insertion of DFT at MQNI determined by this algorithm.**

Keywords—DFT, SCOAP, Testability, Controllability, Observability

I. Introduction

At the time of manufacturing of chips, there is a huge possibility of occurrence of several faults in the chips. Fault detection at later stages of design life cycle is very costlier. Hence the testability of the circuit should be considered at every stage of the circuit design process i.e. from the synthesis stage to the manufacturing stage. To quantify the fault detecting ability of the design, testability measurement techniques are used which indicate how easy or hard it is to test a circuit. The design for testability in digital circuits and the guidance for automated test pattern generators rely heavily on testability measures, which help to reduce both, the test generation complexity and the testing time.

As the digital circuits become larger, the complexities involved of testing process increase and hence the significance of testable design, efficient testability analysis increases. Many times the testability analysis approaches are mentioned in terms of controllability and observability. But as the complexity of combinational circuit increases, calculations for testability by using mathematical formulae become very tedious.

The idea of calculating the controllability and observability and their effect over Design For Testability (DFT) have been discussed in literature available till date. A new concept of sensitivity is introduced to indicate the degree of testability improvement and the global r-modification problem is introduced [8]. The global r-modification deals with making r transformations to a circuit in order to improve its testability. Testability requirement about test parameters such as controllability and observability has been explained [9]. Ruthan's System Model [10] was used for calculating the controllability and observability measures.

In this paper, we present an algorithm which gives the list of Most Qualified Nodes for Improvement in testability (MQNI) and Testability Improvement Factor (TIF)after inserting DFT at MQNI considering area overhead, using a gate level circuit description. A software tool is developed using C language to implement this algorithm. Readily available netlist files (written in verilog language) of ISCAS' 89 benchmark circuits are used as an input to the algorithm.

This paper is organized as follows: Section 2 briefly discusses about concept of DFT, DFT insertion techniques, an overview of testability measures and testability analysis methods. Proposed algorithm and calculations for the testability improvement factor, area overhead will be explained in Section 3. The implementation results will be presented in Section 4 and finally conclusion will be given in Section 5.

II. DFT and Testability Measures

A. Design for Testability

Design for testability (DFT) is a methodology, or a collection of methodologies, which results in the creation of a testable design. DFT refers to additional design efforts which are used to improve testability of the nodes. DFT represents a set of rules that needs to be followed while designing circuit to achieve good testability. One of the DFT techniques is using ad hoc method which is circuit specific. DFT using scan chain methodology is implemented to test boundary as well as internal nodes. To insert the DFT efficiently, it is necessary to have prior knowledge of testability of each node in the circuit. The testability analysis provides brief idea about complexity of the circuit. The DFT is inserted in the circuit to reduce the circuit's complexity.

B. *Testability measures and analysis methods*

While designing the digital circuits, testability is one of the most important measures, along with other measures such as performance and cost. Poor testability of a circuit causes reduction in test quality. High test costs of complex circuits can be from 10% to 70% from whole design time [11]. Hence, it is necessary to improve the testability of device without increase in number of pins and area overhead. It is necessary to consider the testability analysis of the circuit to ensure maximum effectiveness of DFT. The testability can be calculated on the basis of measures of controllability and observability of any node in circuit. Controllability indicates how easily an internal node of a circuit can be set to 0 or 1 by setting the primary inputs. Similarly, observability indicates how easily the state of an internal node of a circuit can be observed from the primary outputs. Controllability and observability obtained in the testability analysis of a circuit can be used for detecting test nodes at which DFT can be inserted to improve circuit's testability and improved testability leads to higher fault coverage.

For determining testability measures there are many methods are available till date [1-7]. Most of the times testability measures like SCOAP [1], CAMELOT [2], and VICTOR [3] are used for testability analysis.

C. *SCOAP*

SCOAP is testability analysis method which gives the testability of different nodes in the circuit based on the different formulae [1]. Combinational 0-Controllability (CC0), Combinational 1-Controllability (CC1) and Combinational Observability (CO) can be considered as the SCOAP values. CC0 is difficulty in setting the node to 0 from primary inputs. CC1 is difficulty in setting the node to 1 from primary inputs. CO is difficulty in observing the node at primary output. The SCOAP values are in the form of gray scale i.e. the values can range from 0 to infinity. SCOAP values represent the difficulty in testing a node. Higher the SCOAP values for the node; it is more difficult to test.

III. Proposed Algorithm

This algorithm accepts netlist of a circuit written in verilog language as an input and suggests best suited nodes (MQNI) to insert DFT. The algorithm is divided into two parts. In the first part, testability of a circuit is computed using SCOAP. In the second part, MQNI are identified to insert DFT. TIF and area overhead are also calculated.

Algorithm Part 1: In this part of algorithm testability of a circuit is computed for each node. The testability analysis is done by using SCOAP method without realization of actual schematic.

Algorithm Part 2: In this part of the algorithm two criteria are used to short list the nodes further selection of MQNI. The first criterion is total testability i.e. addition of CC0, CC1 and CO of a node and the second criterion is fan-out of that node. Total testability should be considered because it gives difficulty in controlling and observing the specific node. Insertion of DFT at nodes having higher fan-out will lead to improvement in testability.

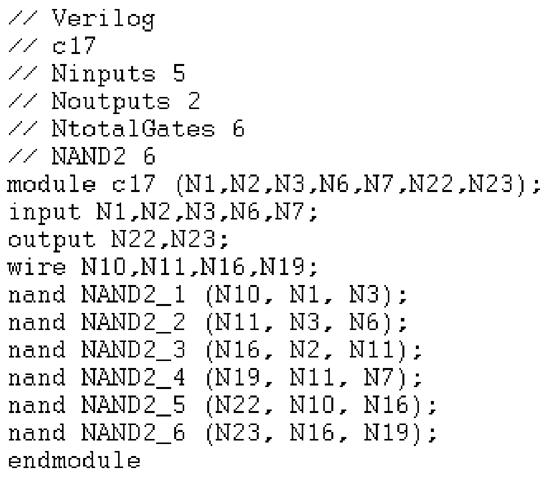
These criteria are applied only on the internal nodes excluding the primary inputs and primary outputs. These values are used to obtain threshold for total testability and fan-out. The nodes having values of total testability (TT) and fan-out (FO) greater than the respective thresholds are shortlisted. Separate shortlist if prepared by considering one criterion at a time. Both shortlists are sorted according to the descending order of corresponding values i.e. testability or fan-out. From these two shortlists, one final shortlist is prepared by giving higher priority to the nodes in the short list of total testability. Hence algorithm selects top-most node from TT and then selects top-most node from shortlist of FO. Repetition of any node in final shortlist is avoided. In this way, both shortlists are merged to form a final shortlist. The nodes in this list are called MQNI. DFT is inserted at these nodes. Testability values after insertion of DFT are compared with the testability values before insertion of DFT. On the basis of formulae given further in algorithm part 2, testability improvement factor is calculated.

A. *Algorithm Part I: Computation of Testability*

In the first part, the netlist file is given as an input to the algorithm and gate specifications are extracted. The netlist gives the information about number of primary inputs & primary outputs, list of primary inputs & primary outputs, list of intermediate wire nodes and the list of gates, inputs to a gate and the corresponding output of it. After extracting the gate description from the file, the SCOAP formulae are applied on the circuit under consideration without realizing its schematic. The computed CC0, CC1, CO values are considered as an outcome of this part of algorithm.

Example: Netlist in verilog language of C17 combinational benchmark circuit from ISCAS'85 series.

The verilog file of C17 is shown in the figure 1. Figure 2 shows the flowchart of the algorithm part1.



```
// Verilog
// c17
// Ninputs 5
// Noutputs 2
// NtotalGates 6
// NAND2 6
module c17 (N1,N2,N3,N6,N7,N22,N23);
input N1,N2,N3,N6,N7;
output N22,N23;
wire N10,N11,N16,N19;
nand NAND2_1 (N10, N1, N3);
nand NAND2_2 (N11, N3, N6);
nand NAND2_3 (N16, N2, N11);
nand NAND2_4 (N19, N11, N7);
nand NAND2_5 (N22, N10, N16);
nand NAND2_6 (N23, N16, N19);
endmodule
```

Figure 1 Sample netlist of C17

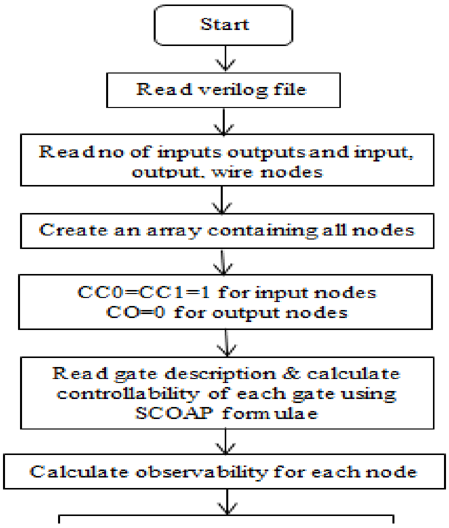

Start
Read verilog file
Read no of inputs outputs and input, output, wire nodes
Create an array containing all nodes
CC0=CC1=1 for input nodes
CO=0 for output nodes
Read gate description & calculate controllability of each gate using SCOAP formulae
Calculate observability for each node

Figure 2 Flowchart for algorithm part 1

Testability analysis on C17 using algorithm part 1 is given in figure 3. As shown in the figure 3, according to SCOAP method and, for primary input nodes N1, N2, N3, N6 and N7, the values of CC0 and CC1 are set to 1 and for primary output nodes N22 and N23, the value of CO is set to 0 and the steps shown in the flowchart are followed.

Figure 4 shows the testability values of C17 calculated manually and these values for each node are verified using generated tool.

Node	CC0	CC1	CO
N1	1	1	5
N2	1	1	6
N3	1	1	5
N6	1	1	7
N7	1	1	6
N10	3	2	3
N11	3	2	5
N16	4	2	3
N19	4	2	3
N22	5	4	0
N23	5	5	0

Figure 3 Sample netlist of C17

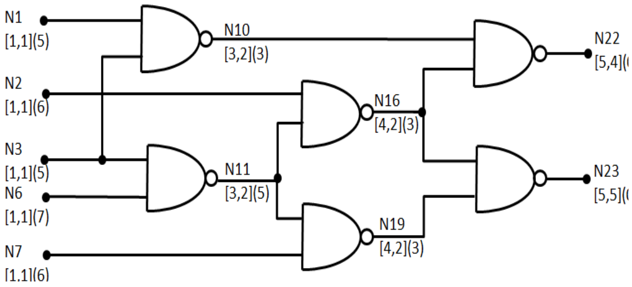


Figure 4 Sample netlist of C17

B. Algorithm Part II: Finding MQNI and computing TIF

In this algorithm most qualified nodes for improvement in testability are identified and testability improvement factor is computed by considering changes in the testability values of all nodes. This part uses testability values obtained from outcome of part 1 of the algorithm. As greater testability values for a node indicate difficulty in testing the node, total testability is considered as one of the criterion. Fan-out of a node is considered as another selection criterion because change in the testability values for a node having more fan out can affect the testability values of other nodes along its branches. These are the selection criteria for determining MQNI. Total testability (TT) is the addition of CC0, CC1 and CO as shown in equation 1. Values of CC0, CC1 and CO are obtained from algorithm part 1 using SCOAP method. Fan-out of a node is obtained by observing appearance of that node as an input for various gates.

Nodes having TT and FO values greater than certain threshold values are further shortlisted for identification process of MQNI. Two shortlists are formed after this stage. By merging of these two lists, a final list of nodes is obtained. These nodes identified as MQNI in testability.

$$\mathrm{TT} = \mathrm{CC0}\,(x) + \mathrm{CC1}\,(x) + \mathrm{CO}\,(x) \qquad (1)$$

Using above formula, different values of TT are obtained for different nodes. From these values, maximum value of TT and minimum values TT as well as maximum value of FO are considered for calculation of thresholds. Thresholds are calculated using the equations 2 and 3.

$$\text{Threshold for TT} = \frac{(\mathrm{TTmax} + \mathrm{TTmin})}{2} \qquad (2)$$

$$\text{Threshold for Fanout} = \frac{\text{maximum fanout}}{2} \qquad (3)$$

Nodes having TT and FO values greater than threshold are included in two separate lists. After preparing two lists, they are sorting in descending order to prioritize the shortlisted nodes. Now a final shortlist is formed by picking one node from each list. MQNI are obtained from this shortlist only. Here, node 11 is common in both the lists. Hence it is not repeated. For C17 circuit, sorted shortlists and final list are as follows:

Shortlist for total testability: Nodes 11, 16, 19.

Shortlist for fan-out: Nodes 11, 16.

Final List: 11, 16, 19.

DFT is inserted at these nodes. Insertion of DFT at a node makes it directly controllable and observable. Thus CC0 and CC1 values for that node become same as that of primary input and CO value of that node becomes same as that of primary output. With these changes in the CC0, CC1 and CO for shortlisted nodes, testability values for other nodes are calculated again. Comparison between original and new testability values after insertion of DFTs is done on the basis of following formulae. The improvement in testability is indicated by testability improvement factor (TIF). All the intermediate nodes are mutually dependent. Therefore insertion of DFT at certain nodes lowers down SCOAP values of other nodes i.e. improves their testability while it may increase SCOAP values of few nodes i.e. decreases their testability. Hence we have proposed the TIF considering both the cases giving equal importance to variation in testability of every node. The second part of this algorithm is implemented using flow chart as shown in figure 5.

Testability improvement for any node x is calculated using equation 4.

$$I(x) = \frac{TTorg - TTnew}{TTorg} \qquad (4)$$

Where I(x) is calculated for every node and x represents node within the circuit. I(x) gives the deviation of testability value from original values. Testability improvement factor (TIF) for all nodes is calculated using equation 5. TIF is summation of I(x) for all nodes within a circuit.

$$TIF = \sum_{\text{all nodes}}^{x} I(x) \qquad (5)$$

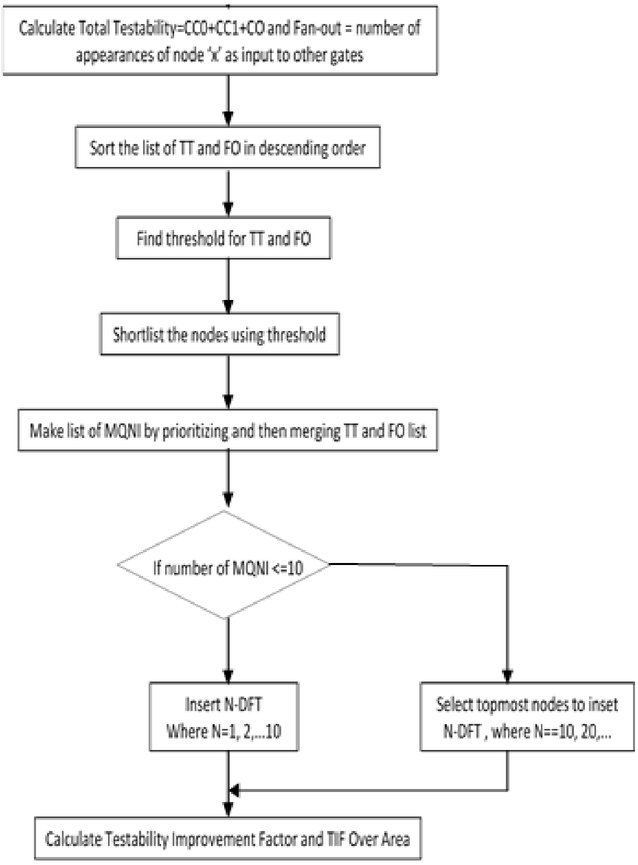

Calculate Total Testability=CC0+CC1+CO and Fan-out = number of appearances of node 'x' as input to other gates
Sort the list of TT and FO in descending order
Find threshold for TT and FO
Shortlist the nodes using threshold
Make list of MQNI by prioritizing and then merging TT and FO list
If number of MQNI <=10
Insert N-DFT Where N=1, 2,...10
Select topmost nodes to inset N-DFT , where N==10, 20,...
Calculate Testability Improvement Factor and TIF Over Area

Figure 5 Flow chart of algorithm part 2

In cadence tool, area is specified in terms of number of cells. Using this information about area of a circuit, percentage area overhead (AO) after insertion of DFT is calculated using equation 6.

$$AO = \frac{\text{no of DFT added} * \text{no of cells required to add 1 DFT}}{\text{no of cells in original circuit}} * 100 \qquad (6)$$

IV. Implementation Results

Results of this algorithm are verified by using some benchmark circuits such as C17, C432, C499 and C880. The improvement in TIF is calculated by inserting DFT at 5 nodes and 7 nodes selected from the list of MQNI. Improvement in testability is greater when DFT is inserted at 7 nodes. But simultaneously area overhead increases which is shown in the result tables I and II. These results indicate that there is a trade of between TIF and area overhead. It means improvement in the testability is achieved at the cost of increased area overhead.

TABLE I: Results for 5 DFT Insertions

Circuit	Total no of nodes	No of MQNI	TIF	%area overhead
C432	243	60	39.842	8.771
C499	243	67	155.436	8.620
C880	443	141	27.336	4.64

TABLE II: Results for 7 DFT Insertions

Circuit	Total no of nodes	No of MQNI	TIF	%area overhead
C432	243	60	51.101	12.28
C499	243	67	163.921	12.06
C880	443	141	36.922	6.50

V. Conclusion

In this paper an algorithm for computation of testability of the combinational circuit and determination of MQNI to achieve effective insertion of DFT is presented. The testability can be improved significantly at the nodes suggested by presented algorithm. A new factor for indication of improvement in testability is introduced which is formulated by considering variation in testability of all the nodes present in the circuit. There is tradeoff between testability improvement factor and area overhead. Therefore the optimum value of both should be considered. Based on this algorithm a tool is designed using C language which requires only netlist file of the benchmark circuit as an input. This algorithm is independent of actual realization of circuit schematic.

References

[1] H. H. Lo, W. F. Lee, M. B. I. Reaz, Member, N. Hisham, A. Y. M. Shakaff, "Design Methodology to Achieve Good Testability of VLSI Chips: An Industrial Perspective",IEEE, International conference on Electronic Design, , 2008

[2] Deepak K.G., Robinson Reyna, Virendra Singh, and Adit D. Singh, “Leveraging Partially Enhanced Scan for Improved Observability in Delay Fault Testing”,IEEE, Asian Test Symposium, 2009

[3] Goldstein L.H. (1979),”Controllability/ observability analysis of digital circuits”, *IEEE Trans. Circuit.* Syst., CAS-26, No.9, pp. 685-693.

[4] Zoran Stamenkovi, Senior Member, Mary Giles, and Francisco Russi “Combining Internal Scan Chains And Boundary Scan Register: A Case Study”, IEEE, 25th International Conference on VLSI Design, 2012

[5] J. E. Stephenson and J. Grason,”A Testability Measure for Register Transfer Level Digital Circuits”, in Proc. Int. Symposium. Fault Tolerant Computing (FTCS), Pittsburgh, PA, June 1976, pp.101-107.

[6] J. Grason, “TMEAS-a Testability Measurement Program”, in Proc. IEEE/ACM Design Automation Conf., 1979, pp. 156-161.

[7] R. G. Bennetts, C. M. Maunder, and G. D. Robinson, “CAMELOT: A Computer-aided Measure for Logic Testability”, IEE Proc. Pt. E, vol. 128, no. 5, pp. 177-189, 1981.

[8] R. G. Bennetts,”Design of Testable Logic Circuits”, Addison-Wesley, Reading, MA, 1984.

[9] Miron Abramovici, Melvin A. Breuer & A. D. Friedman, Digital System Testing And Testable Design, Jaico Publishing House, 2006.

[10] C.P. Ravikumar Gurjeet S. Saund Nitin Agrawal,”A STAFAN-like Functional Testability Measure for Register-Level Circuits”, IEEE test symposium, pp. 192-198, 1995.

[11] H. Joshi. HISCOAP, “A Hierarchical implementation of SCOAP testability nzemure for VLSI circuits”, Master's thesis, IIT Delhi, Dec. 1993.