

Study of VICTOR: A Testability Analysis Algorithm

Aditya Newalkar, *Student Member, IEEE*

Abstract—VICTOR is a testability analysis algorithm with linear complexity. The algorithm assigns controllability triplet and observability triplet - label, weight and size, to every node. The algorithm also defines two operations - select and merge, for assigning the controllability and observability measures for every node in the circuit. The node label is a symbol indicating the signal dependencies at that point. The node weight is assigned to estimate the chance of reconvergence of fanout branches at that node. The node size signifies the number of tests for controlling or observing the fault at that node. VICTOR is a useful tool in addressing acute problems in test generation for large circuits: minimizing the effort in test generation by removing the redundancy in the circuit design, increasing speed of the testability algorithm.

Index Terms—Fast Testability Analysis, VICTOR, Redundancy

I. INTRODUCTION

TESTABILITY analysis quantifies the aspects of testing of electronic circuit, such as: determination of fault coverage depending on the number of test vectors applied, identification of difficult to test faults, redundancy removal and analysis of the efficacy of the *design for testability* (DFT) method. The testability analysis is therefore a useful tool while testing the circuits. Following two qualities are desired for any testability analysis tool:

- The goal is not to generate the test vectors in itself but to analyze the circuit and to aid the test vector generation,
- The algorithm has linear complexity in order to be useful against the *automatic test pattern generator* (ATPG) tools [1].

The testability analysis methods described in [2] [3] predict the parameters of testability of the circuit, based on the *controllability* and *observability* of the constituent signals. The controllability and the observability are defined as “*controllability* refers to the primary input conditions needed to provoke a fault i.e. to set or reset a node” and “*observability* describes the primary input conditions required to propagate a fault to the primary output” [2]. The controllability and observability are numbers indicating the relative effort required to set or reset particular input or output. One of the underlying assumptions while calculating the controllability and observability measures for any signal is that the event of setting or resetting of that signal is independent of any other setting or resetting of any other signal in the circuit.

Manuscript received April 23, 2004. This work was done as a term paper required for ELEC 7250 course conducted by Dr. Vishwani Agrawal in Spring 2004 at ECE department, Auburn University. Peer reviewer is Mr. Kalpesh Shetye.

A. Newalkar is a graduate student with the Electrical and Computer Engineering, Auburn University, USA (e-mail: newalaa@auburn.edu).

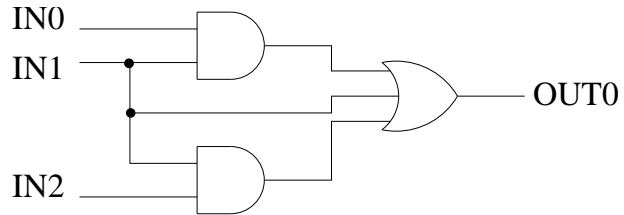


Fig. 1. Example of Reconvergent Fanouts [1]

When a logic signal splits into multiple branches (fanout) and later reconverges into two or more inputs of a gate, the net thus formed is referred as the reconvergent fanout. The testability measures for the signals in the reconvergent fanout stem can no longer be calculated separately. Some of the earlier testability analysis algorithms like SCOPE ignore the case of reconvergent fanout as a special case and treat the fanout stems as independent signals for calculating the testability measures. Therefore the results produced are incorrect. The reconvergent fanouts depicted in figure 1.

An accurate testability analysis algorithm identifies the redundant faults. The redundant faults in the circuit accounts for wastage of as high as 90% of the test generation effort. Testability analysis tools can be useful in identifying and removing the redundancy, thus saving considerable test generation effort.

Thus to make the accurate prediction of testability, the case of reconvergent fanouts must be taken into account while performing the testability analysis. [4] proves that the signal dependencies due to reconvergent fanout signals are responsible for single-redundant faults, under certain conditions. In this report we borrow the nomenclature from [4] for the fanout branches as follows: every fanout branch will be referred by *root-pin-output*. Root is the fanout signal, pin is the number of the input pin and output is the output node.

The study paper is organized as follows: Section I gives general introduction to testability analysis and desirable characteristics of a testability analysis tool, Section II defines the controllability and observability triplets used by VICTOR and demonstrates the use, Section III outlines the working of the algorithm and finally we conclude with the advantages and the characteristics of the VICTOR.

II. TESTABILITY TRIPLETS: EXPLANATION AND USAGE

The circuit under test is modeled as a network of cells, connected by unidirectional links. Each link to a cell represents a fault cite. The single stuck-at fault model is chosen for the analysis of the faults in the circuit. As stated earlier, VICTOR calculates controllability and observability

TABLE I
FOUR VALUE LOGIC

0	logic 0 assignment
1	logic 1 assignment
X	no assignment (don't care)
C	conflicting logic 0 and logic 1 assignment (clash)

triplets viz. label, size and weight for every node. A four-valued logic is used for the calculations is shown in Table II

A. Label Assignment

A Label is a list of logic symbols that represent possible signal assignment for detecting some of the faults in the circuit. Therefore there are *set label* and *reset label* to force the logic value '1' and logic value '0' to sensitize s-a-0 and s-a-1 fault, respectively. The set and reset labels are referred by the subscript 's' and 'r', respectively henceforth.

Invoking the fault alone, is not sufficient but the signal should be propagated at the output. Therefore, *monitor label* is defined by the number of tests to sensitize the path from the node to the primary output.

The labels are evaluated using *select* and *merge* operations to identify the potential redundancy. If the result of either of the operations is a clash, then it indicates a redundancy in the circuit.

B. Size and Weight Assignment

The integer size indicates risk of signal reconvergence, and integer weight indicates total number of patterns existing per node, at that node. Similar to sub-groups in the label testability measure, there are set size and set weight, reset size and reset weight, and monitor size and monitor weight as the sub-groups of testability measures in size and weight group. The signal redundancy in the circuit may set/reset the node V to logic 1/0. The risk of setting/resetting of the node is assessed by set/reset weight. The monitor weight assesses the risk of propagating the value of the signal set by the signal redundancy to the POs. The set/reset size is the number of patterns that set/reset the node. The monitor size is the number of patterns that allow propagation of the fault to the POs.

C. Select Operation

The select operation performs heuristic selection of the label, risk and size parameters of the input/output node to propagate the label, risk and size parameters to output/input. The direction of propagation depends on if it is a controllability calculation or observability calculation. In the former case, the direction of propagation is from input to output and the direction is reversed for the observability calculation.

The select operation always selects the pattern with the lowest risk. Thus by selecting few measurements the algorithm does not run at the risk of selecting the redundant path in the circuit.

TABLE II
FOUR VALUED LOGIC ALGEBRA USED FOR MERGE OPERATION

	0	1	X	C
0	0	C	0	C
1	C	1	1	C
X	0	1	X	C
C	C	C	C	C

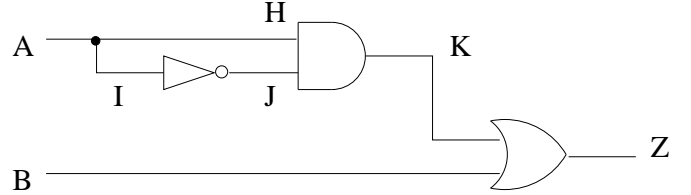


Fig. 2. Circuit Used to Illustrate the Operation of VICTOR [4]

D. Merge Operation

Merge operation executed when performing two operations simultaneously e.g. setting the input value and observing the output value. The labels are merged according to the four valued logic algebra shown in the table II.

III. OPERATION

In this section we elaborate the operational details of the algorithm. VICTOR needs to make three passes over the circuit under test before calculating the test vectors and identifying the redundancy in the circuit. In the first pass circuit is levelized. In the second and third pass controllability and observability measures are respectively calculated. The circuit used to illustrate the operation of VICTOR is shown in the figure 2.

A. Levelization

VICTOR assigns levels to each node according to its distance in number of gates from the PIs. VICTOR then sorts the nodes according to the levels.

- level 0: A, B
- level 1: J
- level 2: K
- level 3: Z

B. Calculating Controllability and Observability

The controllability and observability calculations are similar. In the controllability calculation for each node label, weight and size is determined for each level from lower to higher. In the observability calculation direction of traversal is from higher level to lower level. The labels are initialized as don't care, weights are initialized as zero and size is set to one. For the example shown in figure 2, PI labels are [4]:

$$A_s=1X, A_r=0X,$$

$$B_s=X1, B_r=X0,$$

For fanout nodes, as the risk of reconvergence grows the product of the fanout and the level count to the farthest

PO is assigned to weight.

$$\text{weight}(A_s) = \text{weight}(A_r) = 2*3 = 6,$$

$$\text{weight}(B_s) = \text{weight}(B_r)=1$$

As there is only one way the PI node can be set or reset,

$$\text{size}(A_s) = \text{size}(A_r) = 1,$$

$$\text{size}(B_s) = \text{size}(B_r)=1$$

After the PIs are initialized the the controllability triplets are evaluated for every node and from lower to higher level. Depending on the gate following select and merge operations are applied [4]:

NOT gate:

$$J_s = A_r,$$

$$J_r = A_s$$

AND gate:

$$K_s = \text{merge}(A_s, J_s),$$

$$K_r = \text{select}(A_r, J_r)$$

OR gate:

$$Z_s = \text{select}(K_s, B_s),$$

$$Z_r = \text{merge}(K_r, B_r)$$

Therefore, finally controllability triplets for Z are $Z_T S=X1,1,1$ and $Z_T R=00,7,2$ [4]

For the observability calculations the order of processing of the nodes is OR, AND and NOT, which is reverse that of controllability calculation. The sequence of select and merge operations is applied as shown bellow [4]. Note the name A-1- K_x follows the root-pin-output nomenclature.

OR gate:

$$K_m = \text{merge}(Z_m, B_r),$$

$$B_m = \text{merge}(K_r, B_r)$$

AND gate:

$$A-1-K_m = \text{merge}(K_m, J_s),$$

$$J_m = \text{merge}(K_m, A-1-K_s)$$

NOT gate:

$$\text{merge}(A-1-K_s, A-1-K_m) = CX;$$

$$\text{merge}(A-1-K_r, A-1-K_m) = 00;$$

C. Handling Reconvergent Fanouts

When calculating controllability triplets, if a circuit contains reconvergent fanouts that converge into single output, only one node that is at the lowest level among all inputs is used by the select operation. Therefore, the reconvergent fanouts can be treated as independent cases.

When calculating the observability, the reconvergent fanouts are accounted for in the merge operation. If the merge operation results in a clash then a redundancy due to reconvergence is found.

IV. CONCLUSIONS

A fast testability analysis algorithm is examined. The algorithm is most useful in identifying the redundancy and handling errors in the testability measurement caused by the reconvergent fanouts. The algorithm has linear complexity which is useful when examining large circuits. Most striking attribute of the program as compared to D-Algorithm is that there are no backtracks. This is possible because of the partitioning of the circuit into different levels. As pointed out in some of the later works like [3], the program does not attempt to predict the test coverage.

REFERENCES

- [1] M. L. Bushnell and V. D. Agrawal, *Essentials of Electronic Testing for Digital, Memory & Mixed-Signal VLSI Circuits*, chapter Testability Measures, pp. 129–154, Kluwer Publications, 2002.
- [2] I. M. Ratiu, “VICTOR: A Fast VLSI Testability Analysis Program,” in *Proceedings of IEEE International Test Conference*, November 1982, pp. 397–401.
- [3] Franc Brglez, “On Testability Analysis of Combinational Networks,” in *Proceedings of the International Symposium on Circuits and Systems*, May 1989, pp. 221–225.
- [4] I. M. Ratiu, *VICTOR: Global Redundancy Identification & Test Generation for VLSI Circuits*, Ph.D. thesis, University of California, Berkeley, CA 94720, May 1983.