International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:10, No:1, 2016

Reducing Test Vectors Count Using Fault Based Optimization Schemes in VLSI Testing

Vinod Kumar Khera, R. K. Sharma, A. K. Gupta

***Abstract*—**Power dissipation increases exponentially during test mode as compared to normal operation of the circuit. In extreme cases, test power is more than twice the power consumed during normal operation mode. Test vector generation scheme is key component in deciding the power hungriness of a circuit during testing. Test vector count and consequent leakage current are functions of test vector generation scheme. Fault based test vector count optimization has been presented in this work. It helps in reducing test vector count and the leakage current. In the presented scheme, test vectors have been reduced by extracting essential child vectors. The scheme has been tested experimentally using stuck at fault models and results ensure the reduction in test vector count.

***Keywords*—**Low power VLSI testing, independent fault, essential faults, test vector reduction.

I. Introduction

THE fundamental objective of the testing is to detect the faults and filter out defective and substandard ICs. The shrinking die space, increasing complexity and enormous clock speeds of application circuit have left behind the Moore's law [1]. This trend has led to the increased susceptibility of faults during manufacturing. Imperfections and defects in the manufacturing process lead to faults and necessitate testing of the manufactured ICs. The response of circuit is evaluated w.r.t. test stimuli generated by Automatic Test Equipment (ATE). For legitimate operation and coveted results, good quality of circuits (reliable) remains the genuine necessity [2]. So, testing plans are basically required for the successful fault recognition and making them more reliable. This will ensure improved reliability of the manufactured circuits. With the increasing integration, test time and hence test cost is increasing rapidly [3]-[6]. Modern Automatic Test Pattern Generation (ATPG) tools tend to reduce the test vector count at the cost of increased intra-vector as well as inter-vector switching activities. It results in increased test cost as well as total test power. Likewise, the decrease of test vectors with the help of don't care less bits can't be done while utilizing random test vector generation scheme. Circuits with Design for Testability (DFT) features, have more number of concurrent active blocks resulting in increased test power in test mode. These limitations necessitate a technique that generates reduced number of test vectors and reduces inter-vector and intra-vector switching as well. Test cost reduction can be realized by using optimized set of test vectors [7]. Static compaction scheme is proposed by [6] in which vectors are compacted after generation. Dynamic compaction scheme as proposed by [8] generates vectors by constraints of previously generated vectors and it is able to detect large number of faults [7], [8]. M. H. Schulz [9] proposed Reverse-order fault simulation technique (ROF), which fails to eliminate redundant test cases. Fault based test vector reduction is also proposed by identifying essential and independent faults. Two faults are said to be independent if they are not detected by a single vector [10], [11]. A fault is said to be essential fault if it is detected by only single test vector in test set. Independent fault based and essential fault based techniques have been proposed by [10], [12]-[14]. Compaction techniques based on independent faults have been proposed in [10], [15]. It is already proven [16] that minimum test set size cannot be smaller than the size of the largest set of independent faults. We have presented techniques for reducing the test vector count based on faults based optimization schemes.

The rest of this paper is explained in detail in below sections. The presented scheme is elucidated in Section II, here section II *A* explains the complete algorithm, Essential fault based optimization (EFBO) scheme is explained in Section II *B* and Independent fault based test vector optimization (IFBO) scheme is explained in Section II *C*. Section III details about removal of redundant test vectors, filling of don't care (X) bits is explained in Section III *A*. Section IV depicts the results obtained.

II. The Proposed Scheme

Test Vector count is the main component in deciding test cost of a circuit, this leads to the motivation to optimize test vectors count considering both independent and essential faults. We proposed heuristic approach for reducing test vector count. The algorithm is as shown below, the technique optimizes test vector count by consideration the combined set of essential and independent faults. The sequence of steps contained in the proposed scheme is shown in Fig. 1

A. Algorithm

- Generation of test vectors
- True Value Simulation (Ti)
- Find Test Vector Detection Count (TVDC) for all faults
- Arrange the faults in ascending order of TVDC
- Find the essential faults

Vinod Kumar Khera is with the Atmel R&D India Pvt. Ltd, Sector-16A, Noida, U.P India (e-mail: vinod_kumar.khera@atmel.com).

Prof. R.K. Sharma and Prof. A.K. Gupta are with the Department of Electronics and Communication Engineering, NIT Kurukshetra, India (e-mail: mail2drrks@gmail.com, anilg699@rediffmail.com).

- Essential Fault based test vector optimization EFBO() //for reducing the intra vector switching and the test vectors*
- Independent Fault based test vector optimization: IFBO() //for reducing inter-vector switching
- Other Fault/ Remaining test vector optimization
- Merge test vectors
- Substitute don't care bits (X) with suitable bit (0/1) using genetic algorithm and hamming distance, so that majority bits are similar // this ensures minimum power dissipation
- Removal of redundant test vectors // Redundant patterns are removed while merging IFBO Based test vectors

* Intra vector switching refers to charging (0 ⟶1) /discharging (1➔ 0)/of parasitic capacitance.

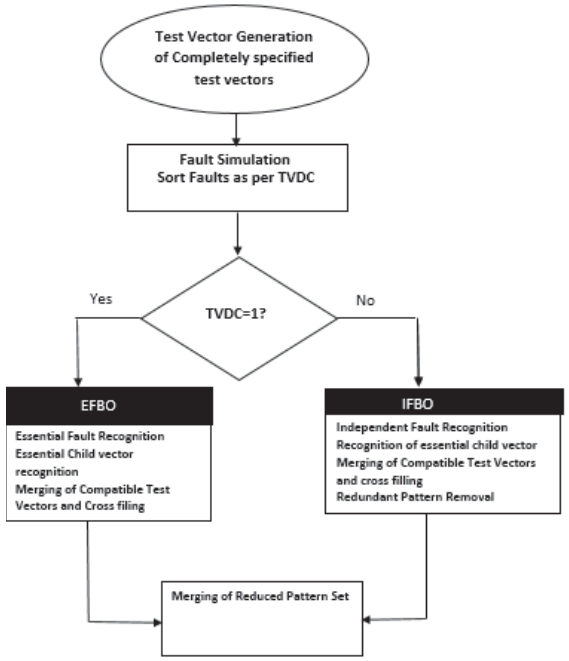
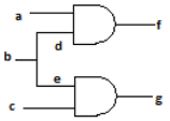


Fig. 1 Test Vector Reduction Scheme

B. Essential Fault Based Test Vector Optimization (EFBO)

Essential fault of a test vector is a fault that is detected only by a unique test vector in the test set [22]. During EFBO phase, we optimize test vector count based on essential faults. This optimized set of test vectors leads to reduced intra-vector switching and curtailing of test power in circuit under test. Essential test vector detects at least one unique fault which is not detected by other test vectors in the test set. In this work, the methods proposed in [14] and [18] have been extended for optimization of essential test vectors. To achieve this, CUD is fault simulated with fully specified set of test vectors generated by ATPG tool. The test vectors are sorted as per their test vector fault detection count (TVDC). Faults with TVDC = 1 are called essential faults. We have used ATLANTA tool for test vector generation and fault simulation. During the whole process of optimization, original fault coverage is preserved. Next, Essential child vectors are generated by relaxing parent test vector. Essential child vectors are the test vectors with optimum number of 1's and 0's and rest of the bits are don't care (X). Let the original set of test vectors generated by an ATPG tool be Ti, and the derived set of essential child test vectors with reduced number of bits be Tc. The test vector set as generated by ATPG is fault simulated to detect number of faults detected by each test vector (TVDC). The detected faults are then sorted in ascending order of their corresponding TVDC count. As discussed earlier that faults with TVDC=1 are essential faults, next step is to find the essential child vector out of these vectors which have TVDC=1. The essential child test vectors are the test vectors with optimum number of optimum don't care bits, but have similar fault coverage as that of parent test vector. To identify maximum don't care bits in a test vector, we have used the technique described in [17]. It spots the primary outputs where there are utmost faults. Next, we locate the path to this primary output from this fault by back traversing and find the values on the lines in between (FAN IN lines). So an essential child test vector may contain a combination of 1s, 0s and don't care (X) bits. Fig. 2 shows the circuit, essential vectors (T_i) and derived essential child vectors (T_{ci}) (where $1 \leq i \leq 4$) respectively.



2(a)

Ti	a	b	c
T1	1	1	1
T2	0	1	0
T3	0	0	0
T4	1	0	1

2 (b)

Ti	a	b	c
T1	X	1	1
T2	0	1	0
T3	0	0	X
T4	1	0	1

2 (c)

Fig. 2 Essential child vector extraction: (a) 3 Input, 2 output circuit; (b) Essential vectors; (c) Essential child vectors

There may be identical test child vectors detecting different faults, such test vectors are compatible. As the objective of any test generation scheme is to reduce the test vectors and achieve maximum fault coverage, the test vectors are checked for compatibility. So that we may merge the compatible test vectors and hence reduce the test vector count. Two test vectors Ti and Tj are called compatible test vectors if and only if none of the columns contain opposite i.e. $T_i = T_j$ where i, j are same column bits of Ti and Tj and ($1 \leq i, j \leq n$). The identified compatible test vectors Ti and Tj can be merged together to a single test vector Tij. We have used the test vector compaction scheme as specified in [16]. The process of finding compatible test vectors is explained in Fig. 3, here T_i and T_{i+1} are successive test vectors. We have elaborated the process of finding compatible vectors in Figs. 3 (a) and (b), respectively.

Ti/Ti+1	0	1	X
0	0	-	0
1	-	1	1
X	0	1	X

Test Vector Ti	Test Vector Ti+1	Compatibility Check
01101X1X	0110X011	Compatible
110X0X10	110X1X10	Incompatible

Fig. 3 Compatibility Check; (a) Compatibility rules; (b) Compatible Vectors

The merged test vector Tij now detects faults detected by both Ti and Tj as shown in Fig. 4. Here, we have shown 4 test vectors, T1, T2, T3 and T4 along with their essential child test vectors. T1, T2 are merged as rules in Fig. 3. This process results in reduced test patterns for a CUD (we have written a fault simulator in C language for these steps). Remaining bits other than those of child vectors are the don't care (X) bits. These X bits are substituted with suitable bit value (0s or 1s) using genetic algorithm [12]. For effective selection of don't care (X) bits, the algorithm proposed by [17] has been used. Next, the essential test vectors obtained are clustered based on compatibility as specified in [18].

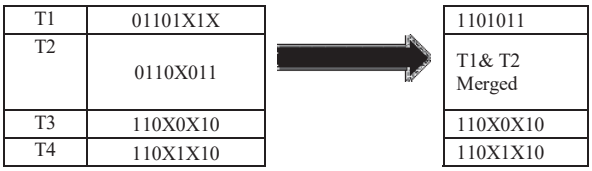


Fig. 4 Merging operation

Essential Fault Based Test Vector Optimization (f, Ti)

For every essential fault of Ti

- justify the Primary inputs (PI) to gate and assign true value to Primary output (PO) of this path
- Find and justify the cone for the line of this fault
- Merge PI values fixed at steps a and b

For every essential child vector T_c (map to compatibility set)

- If there exists no sets; create a new set and map Tc Else-if map the Tc to existing compatible set
- Else create new set and map the Tc to new set // this takes care if no compatible set exists
- Go back to initial Step // repeat until all child vectors are mapped
- Simulate Faults using the set produced by merging compatible sets, drop all those detected and store test patterns in an array

C. Independent Fault Based Test Vectors Optimization:

Two faults are said to be independent if they are not detected by a single test vector [10], [11]. Compatible fault test based compaction is proposed in [10]-[12], [14]. In [10], ordering of target fault set using compatible fault set is proposed, whereas in [11], effectiveness of COMPACTEST resulting in improvements over existing schemes is proposed. Independent fault sets have been used effectively in [10] to derive larger independent fault sets. However, this technique is suitable for smaller circuits only. Further clustering is proposed using independent faults by [14] for reduction of test patterns. Clique utilization for fault clustering is described in [13] to determine the maximum independent fault set. However, [13] is suitable for small network only due to dependency issues. In the proposed technique, all faults with TVDC other than 1 are considered for designing independent graph and independent faults are found as per rules specified in [19]. Authors have modified the algorithm proposed in [14] by applying the genetic algorithm to recognize redundant test vectors. The proposed algorithm has capability to optimize both independent fault based test vectors as well as compatible fault based test vectors and is with added advantage of removal of redundant test vectors. The algorithm for independent fault based optimization is shown below.

Algorithm IFBO

- For every independent fault(s)f that is detected by a set of test vector T
- For every test vector t (where t is a member of T):
    - Find the child vector tc from t (as done earlier for essential vectors)
        - For every child vector T_c (map to compatibility set)
        - If there exist no sets; create a new set and map Tc
            - Else-if map the Tc to existing compatible set
            - Else create new set and map the Tc to new set // this takes care if no compatible set exists
- Go back to initial Step // repeat until all child vectors are mapped
- Simulate Faults using the set produced by merging compatible sets, drop all those detected and store test patterns in an array
- Filling the don't care bits
- Remove the redundant test vectors.

III. Removal of Redundant Test Vectors

To reduce test vector count further, redundant test vectors are found from test vectors obtained after merging the vectors from EFBO and IFBO. The test vectors thus obtained after cross filling are fault simulated, all test vectors are arranged in decreasing order of fault recognized. We have used two variables *one_check* and *check* for recognizing the essential and minimal set of test vectors as in [13]

- Calculate one_check for each test pattern
- Patterns with non-zero one_check are copied to the final set Step
- All faults detected by these patterns are dropped from the fault list and also from the fault list of remaining vectors
- While (there are still remaining faults)

o Find a pattern that detects the largest number of uncovered faults

o Add it to the final set and drop this fault as in Step 3.

A. Filling of X-bits

In this step, don't care (X) bits obtained by merging test vectors of proposed algorithm are replaced with appropriate bit values (1s/0s). This is achieved by using Genetic algorithm proposed in [12].

IV. Results and Observations

Algorithm proposed in this work has been successfully applied on fully specified test pattern sets generated by Atalanta [19]. Standard ISCAS circuits have been tested using Atalanta and the scheme proposed in this work. Fully specified test pattern sets for single stuck at fault model are

generated using ATPG tool Atalanta [19]; Table II shows performance comparison between the proposed scheme and ATPG tool Atalanta [19]. Column 2 and column 3 show test patterns generated by ATalanta with _x option and test patterns generated by proposed method respectively. The reduction (ranging from 15% to 80 %) is shown in column 3 of Table I.

TABLE I
COMPARISON OF TEST VECTORS PROPOSED VS ATALANTA

Circuit	Test Vectors Atalanta [19]	Test Vectors Proposed Scheme	Comparative reduction (%)
C2670	113	96	15.04
C5315	193	61	68.4
c3540	95	63	33.69
s1238	158	117	25.95
s5378f	256	89	65.3
s9234	366	227	37.98
s13207	633	202	68.08
s38584	667	253	62.06
s38417	900	208	76.8
S15850.1	657	99	84.93
s9234	620	195	68.54

TABLE II
% TEST VECTOR COMPACTION OF ROF OVER ATALANTA

Circuit	PIs	Atalanta	ROF [9]	ROF over Atalanta
c2670	233	113	106	6.19
C5315	178	193	119	38.34
c3540	50	95	83	12.63
s1238	32	158	124	21.51
s15850.1	611	657	456	30.59
s5378f	214	256	252	1.59
s9234	247	620	375	39.51
s13207	700	633	476	24.80

TABLE III
% TEST VECTOR COMPACTION OF PROPOSED OVER EXISTING TECHNIQUES

Circuit	PIs	RM	RM over Atalanta	Proposed	Proposed over Atalanta
c2670	233	100	11.50	96	15.04
C5315	178	106	45.07	61	68.39
c3540	50	80	15.78	63	33.68
s1238	32	119	24.68	117	25.94
s15850.1	611	181	72.45	99	84.93
s5378f	214	145	43.35	89	65.23
s9234	247	202	67.41	195	68.54
s13207	700	252	60.18	202	68.08

Table II shows comparison of test vector reduction by preexisting techniques, in terms of comparative %age reduction of test vectors as compared to standard ATPG Tool [19]. Further as a proof of concept we have compared the results with [20]. In Table III, we have compared with the preexisting techniques [21], [19], [14]. Here, test vector count as generated by RM and its comparison with Atalanta is shown in Column 3 and 4. Column 5 shows the test patterns generated by proposed scheme while in the column 6, we have compared the test pattern %age saving when compared with Atalanta [19]. It has been proved that the presented scheme has ability to reduce the test pattern more than 80%

V. CONCLUSION

Design of effective scheme for reduction of test vector count has been presented here. Test vector count plays an important role in deciding the test power. In this work, we have effectively shown the reduction in test vectors as compared to existing methods. The method has been tested on standard ISCAS circuits as mentioned above.

REFERENCES

[1] "International Technology Road Map for Semiconductors," 2013.
[2] Hnatek, E.R., "IC Quality – Where Are We," in Proceedings of the IEEE International Test Conference, pp. 430–445, 1987.
[3] Report, "International Sematech, The International Technology Roadmap for semiconductors (ITRS)," 2001.
[4] Chandra, "Reduction of SOC Test Data Volume, Scan Power and testing time using Alternate run-length Codes," in ACM/IEEE Design Automation Conference (DAC), 2002.
[5] Athas, "Low power digital systems based on adiabatic-switching principles," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 2, no. 4, pp. 398-416, 1994.
[6] M. Abramovici, Digital System Testing and Testable Design, Piscataway- New Jersey: IEEE Press, 1994.
[7] C.S. Lin, "Test Compaction for combinational Circuits," IEEE Transactions on CAD, vol. 14, no. 11, pp. 1370-1378, 1995.
[8] B.C. Rosales, "Test Generation and Dynamic Compaction of Tests," in International Test Conference, 1979.
[9] M.H. Schulz, "Socrates: A highly Efficient Automatic Test Pattern Generation Scheme," IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems, vol. 7, no. 1, pp. 126-137, 1988.
[10] S.B. Akers, "On the Role of Independent Fault Sets in the Generation of Minimal Test Sets," in International Test Conference, 1987.
[11] Pomeranz, "COMPACTEST: A Method to Generate Compact Test Sets for Combinational Circuits," IEEE Transactions on CAD, vol. 12, no. 7, pp. 1040-1049, 1993.
[12] S.Kundu. et.al, "A Novel, Technique to reduce both leakage and peak power during scan testing," in IEEE Region 10 and Third International Conference on Industrial and Information Systems, 2008.
[13] S. K. et.al, "Customizing completely specified pattern set targeting dynamic and leakage power reduction during testing," INTEGRATION, the VLSI Journal, vol. 45, pp. 211-221, 2012.
[14] A. El-Maleh, "Test Vector Decomposition based static compaction based static compaction algorithms for combinational circuits," IET Computer Digit, vol. 1, no. 4, pp. 364-368, 2007.
[15] B. Krishnamurthy, "Test Counting: A tool for VLSI Testing," IEEE Design & Test of Computers, vol. 6, no. 5, pp. 58-77, 1989.
[16] Z. Navabi, Digital System Test and Testable Design using HDL Models and Architectures, London: Springer, 2011.
[17] S. Kajihara, "XID: Don't care Identification for Test Patterns for combinational Circuits," IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems, vol. 23, no. 2, 2004.
[18] V.D. Agrawal, "Diagnostic and Detection of fault collapsing for multiple output circuits," in Design, Automation and Test (DATE), Europe, 2005.
[19] H.DSHA, "Atalanta: An Efficient ATPG for Combinational Circuits," Virginia Poly. Institute and State University, Virginia, 1993.
[20] Al-Suwaiyan, "An efficient test Relaxation Technique for combinational and full-scan sequential circuits," in VLSI Test Symposium, Monterey, CA, 2002.
[21] G.Tromp, "Minimal Test Sets for combinational Circuits," in International Test Conference, 1991.
[22] Hamzaoglu, I.; Patel, J.H., "Test set compaction algorithms for combinational circuits," Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, vol.19, no.8, pp.957-963, Aug 2000