ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijareeie.com

Vol. 6, Issue 11, November 2017

An Optimal Machine Learning approach for Fault Coverage Improvement on Silicon Through Functional Testcases

Anilkumar T. G.[1], Rakshith C. A.[2]

Principal Consultant, Architects Academy (TT), Wipro Ltd., E-City, Bengaluru, India[1]

Independent Researcher, Bengaluru, Karnataka, India[2]

ABSTRACT:The Fault Coverage Improvement is a cumbersome and time-consuming taskin the*Design for Test* activity.As per the Moore's Law the present generation Digital Integrated Circuits are facing the challenge of packing million of logic gates, while the die size is shrinking. This in turn demands the DFT engineers to achieve higher fault coverage through ATPG or BIST techniques that are facilitated through various popular EDA tools from Mentor Graphics, Synopsis, Cadence Inc., Although, with automated ATPG or BIST techniques achieving fault coverage beyond 98%with AC & DC test vectors is difficult due to various technical reasons. The prominent approach is to work on ATPG untestable faults thatwere not covered by scan-chain or BIST. This is becausebythe virtue of the design or sometimes more test cycles required to detect those faults. Hence, as a general practise, on top ofATPG test vectors the Silicon test process also include some of the functional test vectors that are targeted to cover these ATPG untestable faults. And major challenge is to select the potential function test vectors from thousands of functional test cases that can cover these ATPG untestable faults. The selection of the best set of functional test cases can be time consuming, iterative and exhaustive process. Hence in this paper, using the Machine Learning approach, the selection of high yielding functional testcases is illustrated.

KEYWORDS: Machine Learning, Design *For* Test,Silicon Manufacture Defects, Fault Coverage.

I.INTRODUCTION

The end-to-end Silicon chip design process involves in digital design with RTL, functional testing, Design for test using scan chain insertion for ATPG(Automatic Test Pattern Generation) or BIST(Built in Self Test), Physical Design with layout and tape out to Silicon foundry for manufacturing the Integrated Circuits in bulk. These manufactured chips are required to undergo test and separate the good chips from those faulty chips due to manufacture defects. The general practice that is followed is to run the ATPG test vectors on the Silicon Tester set-up like Teradyne, VLCT or any other Silicon testers.The ATPG test vectors that are pre-generated from the netlist which provides the total fault coverage that can be achieved with DC and AC ATPG test patterns. The ATPG AC patterns ensures coverage of both stuck-at-fault and transition faults as they are run at-speed and hence initial coverage is obtained. This initial fault coverage is usually around 85% to 90%. The remaining uncovered faults are targeted with ATPG DC patterns that can further take the fault coverage up to 98%. And the residue faults are attempted with the large pool of functional test vectors. This ATPG untestable faults percentage could vary on several factors in a complex System-on-Chip. The reasons for these untestable faults could be due to Flip-Flops that could not be covered under scan chain due to time critical blocks like cache modules tagged to CPU or any proprietary modules orreused hard macros or uncovered faults on some of the combo logic etc., Also,with the BIST method that uses Pseudo pattern test generation & fault detection process can take long run time to meet the desired fault coverage level. And, in some of the commercial microcontrollers like PIC microcontroller where the die area, pin count constraints, cost are more critical than having Scan-insertion to follow ATPG methods. Hence, such chip manufacturerswho follow the proprietary test methods and fault coverage that is solely dependent on the functional test cases developed. This process involves in running functional testcases at gate level simulation and capturing the logic values at Input & output pins in the form of VCD (value change dump) file. These VCD files are converted in to test patterns in the format as desired by the type of

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijareeie.com

Vol. 6, Issue 11, November 2017

Silicon tester being used.Also, these functional test vectors are fault graded with Fault simulation tool and determines the fault coverage percentage by functional test vectors.
The next challenge is to obtain the maximum fault coverage with the functional test cases of lesser test run time with the constraint of limited tester memory capacity.The cost of testing is significant factor that decides the overall production cost. Apart from number of pins included in testingcost, it is also directly proportional to total test vectors, run time and size of the test vectors. Hence the number of faults detectable for every test vector becomes important on choosing the test vector set that addresses the overall test vector size on tester memory. And total test time per chip is also important especially for the low-cost chip set that are to be sold in million. Also, warranty service can incur huge price when testing quality is not met.Therefore, it is required to have optimum selection of test vectors that ensures maximum fault coverage with less number of test vectors. The total number of functional test cases can be about thousand and each of these functional test vectors may cover majority of faults that are common across multiple test cases. Hence,it is required to have an efficient method to select the high yielding test cases with maximum fault coverage with minimum runtime that can fit inside the available tester memory.The traditional trial and iterative methods are time consuming and hence in this paper we propose optimum Machine Learning method to choose the optimal set of required functional test vectors.

II. METHOD 1: A TRADTIONAL APPROACH

In the traditional method the system level functional test cases are prioritized based on theirrun time. Because, it is considered that the system level functional test cases with larger run time are expected to cover more number of faults. If we have N number of functional test cases and after sorting we get the collection of testcases in order named as T_1, T_2, T_3,..... T_N.Then the fault grading is initiated by loading total number of faults on a fault simulator by exciting with test patterns created by test case T_1that results with certain number of detected faults as F_{1D} + undetected faults as F_{1U}. The residueor undetected faults $F_{1U}$are taken as input fault list for test pattern of T_2that results in F_{2D}+ F_{2U}. This process is continued until all the undetected faults get exhausted or till all the testcases get consumed. During this process,notall of test patterns detect the significant number of faults and hence some of these patterns must be discarded because of less yield. And hence this traditional approach of fault coverage improvement does not guarantee the optimization of neither the test vector size nor the test run time.

III. METHOD 2: SUCESSIVE FILTERING METHOD

This is yet another adhoc method which gives optimum results but at the expense of large iteration run time it takes during successive or bubble filtering fault grading method. In this method, from the N number of functional test cases any potential test case say$T_1$is chosen and fault grading is initiated by loading total number of faultsF_{TU} on a fault simulator by exciting with test pattern T_1that results with certain number of detected faults as F_{1D} + undetected faults as F_{1U}.And next step is to load F_{1U} as input fault list for 2nd round of iteration. In the 2nd round of fault grading iteration the same fault list $F_{1U}$is taken as common input fault list for all the test patterns except T_1. As $T_1$is already considered in 1st round and is moved to potential test pattern basket. The outcome of the 2nd round of fault grading with test pattern fromT_2, T_3 …. T_n is (F_{2D} + F_{2U}), (F_{3D} + F_{3U}), (F_{4D} + F_{4U}), (F_{5D} + F_{5U}) …… (F_{nD} + F_{nU}). The testcase that has detected more faults in 2nd round is considered as next potential test case after T_1. Say in this case, F_{2D} >F_{3D}or F_{4D} orF_{5D} or F_{nD}. So, the test case T_2 that has detected F_{2D} which is the maximum number of faults is considered as 2nd potential pattern. Similarly the process is repeated for 3rd round with F_{2U} serving as common input faults for all the test pattern from T_3 *to* T_n that results in (F_{3D} + F_{3U}), (F_{4D} + F_{4U}), (F_{5D} + F_{5U}) …… (F_{nD} + F_{nU}). And the process is repeatedto identify the F_{xD} which is maximum fault detected from one of the T_3 *to* T_n test pattern say in this case it is F_{3D} that has yielded from T_3 test pattern. This process will be continued till all the undetected faults gets exhausted or till the test cases become obsolete. Hence, in the worst scenario the maximum time to complete the sorting of test pattern is $N!$ units of time. Where N is the total time taken by all test patterns T_1 *to* T_n to complete the fault simulation. The fault simulation time is much higher than the functional simulation run timedone on pre-layout netlist. Because the fault simulator takes 2 rounds in which the 1st round is learning process called as good simulation and in the 2nd round it involves in checking for every set of test vector if any of the faults can be detected from the set of input faultsthat is called as fault simulation and can take more than 10x time of good simulation based on total number of faults loaded along with

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijareeie.com

Vol. 6, Issue 11, November 2017

circuit complexity.Although this method gives an optimum result in sorting and prioritizing of test patterns, but it is considered as an exhaustive and time-consuming process.

IV. METHOD 3: PROPOSED BEST APPROACH WITH MACHINE LEARNING METHOD

In this method we are proposing an efficient automation process in which we can sort out the high yielding test patterns that can reduce the fault simulation time from N! time units to just N time units. To start with all the functional test patterns from T_1 *to* T_n are run with fault simulation by loading same set of total faults $F_{TU.}$ So, the total fault simulation time taken is $\sum_{i=1}^{N} T_n$ which is nothing, but sum of time taken by all test patterns T_1 *to* T_n equal to N units of time. As an outcome of this set of fault simulation running with test patterns from T_1 *to* T_n we get the fault list set ($F_{1D} + F_{1U}$), ($F_{2D} + F_{2U}$), ($F_{3D} + F_{3U}$), ($F_{4D} + F_{4U}$), ($F_{5D} + F_{5U}$) …… ($F_{nD} + F_{nU}$). Considering that the test pattern that has resulted in maximum fault detection say F_{1D} from test pattern T_1 Then take the undetected faults list of test pattern T_1 that is F_{1U}which is our potential undetected fault list to be covered by identifying potential high yielding test pattern to be chosen from $T_2$to $T_{N.}$ Unlike in the previous method we do not perform the successive fault simulation in multiple iterations which was found to be exhaustive but very time-consuming method.
A python script is developed that can smartly identify the potential high yielding test patterns without the need of further fault simulation iterations. The logic behind the machine learning script written in python is to compare the fault list of $F_{1U}$with $F_{2D,}$ $F_{3D,}$ $F_{4D,}$ F_{5D} ....F_{nD} which is done by (F_{1U}~ F_{2D}), (F_{1U}~ F_{3D}), (F_{1U}~ F_{4D}), …(F_{1U}~ F_{nD}). With this process, the set that results in lesser number of fault number is considered as next potential test pattern say in this case say (F_{1U}~ F_{2D}) =F_{U2} =>*min*whencompared to any other set. The logic statement for sorting in python is written as follows.

```
>>>list(set(F1U)- set(F2D))
#that results in FU2min.
```

Where $F_{U2min.}$ is considered as fault list equivalent to successive fault grading result of $T_{1\ and}$ $T_{2..}$
The process is continued with F_{U2} fault list compared with F_{3D} ,F_{4D},F_{5D},F_{6D}&F_{nD} which is done by (F_{U2}~ F_{3D}), (F_{U2}~ F_{4D}), …(F_{U2}~ F_{nD}). With this process the set that results in lesser number of fault number is considered as next potential test pattern say in this case say (F_{U2}~ F_{3D}) = F_{U3} =>*min* when compared to remaining other set.So, logic statement in python is as shown below

```
>>>list(set(FU2)- set(F3D))
#that results in FU3min.
```

The machine learning and sorting process through python script continues till F_{Ux} becomes null/least value or till the stage where the F_{Ux} -F_{Ux-1} = 0 which means we have reached the saturation stage and hence discard the rest of the test patterns that are redundant.Hence, without need of successive fault grading we can sort out and identify high yielding test patterns that would go on tester for production test purpose.

V. RESULT AND DISCUSSION

From the Fig 1a& 1b,with *Method 1* it is evident that in the worst scenario, the test vector size can be maximum which is equal to sum of test pattern sizes. But the fault simulation of all test patterns stands at minimum time units.

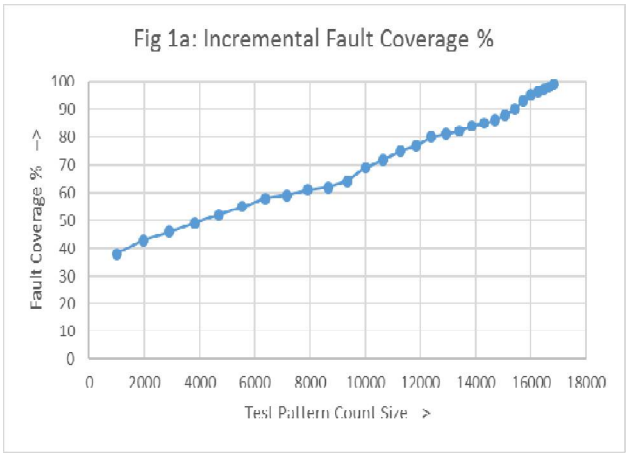
ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

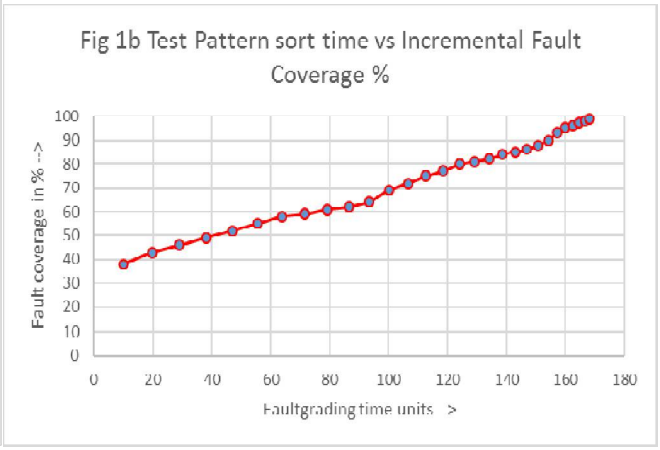
International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

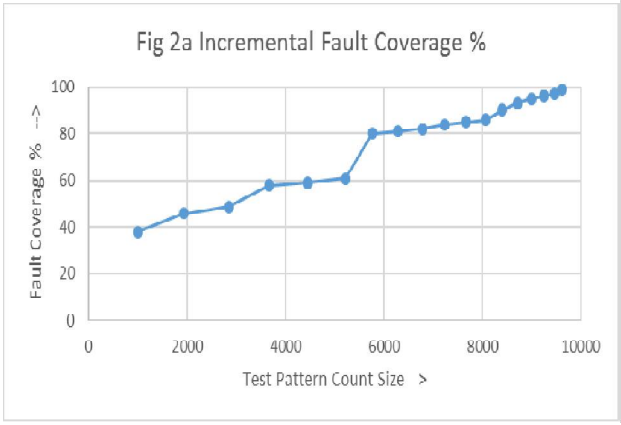
Website: www.ijareeie.com

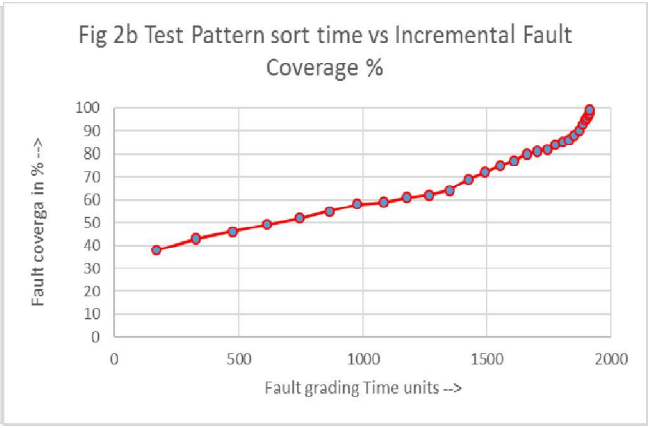
Vol. 6, Issue 11, November 2017



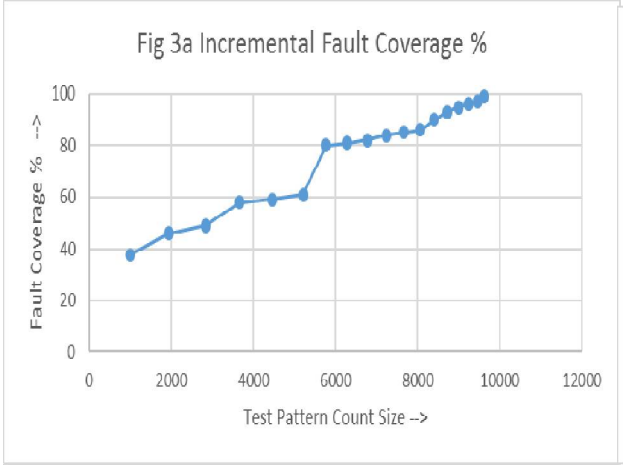


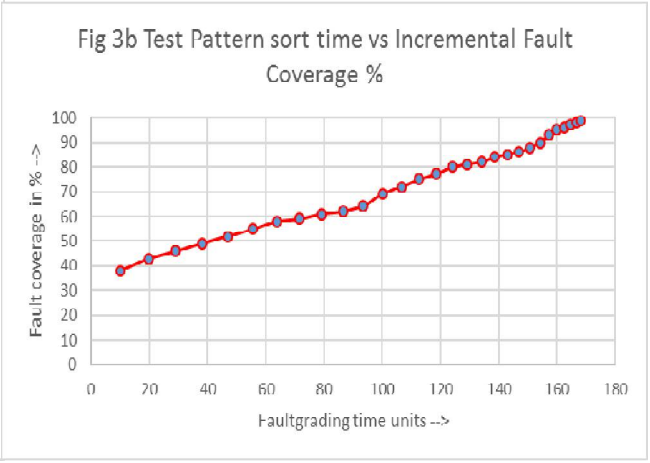
From the Fig 2a & 2b,with *Method 2* it is evident that the optimum test vector size can be achieved but the fault simulation and sorting of all test patterns time is maximum (factorial of N time units--> N!)





From the Fig 3a & 3b,with *Method 3* it is evident that the optimum the test vector size can be achieved same as *Method 2* and the fault simulation and sorting of all test patterns time is also minimum (just equal to N time units)





ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijareeie.com

Vol. 6, Issue 11, November 2017

VI.CONCLUSION

As can be seen from the graphs of fault coverage vs test vectors obtained from Method1, Method2 and proposed machine learning Method3 that our proposed method results with high yielding test vectors that are minimum in size with highest fault coverage when compared to Method1. Although the Method2 produces same results as Method 3 but it takes *N!* time units for fault grading of test cases where as the Method3 takes only *N* time units. Hence our proposed method saves significant time& cost for both test vector processing task by DFT team and for the PE team who work on Silicon testers.Further scope of enhancement on Method3 can be done with the machine learning algorithms viz., decision tree algorithm can be applied for selection of test patterns considering multiple factors like power consumption, average fault coverage per test vector and for any other significant test coverage parameters that influences fault coverage activity.

REFERENCES

[1] Nagi, A. Chatterjee and J. A. Abraham, "Fault simulation of linear analog circuits," Journal of Electronic Testing: Theoryand ApplicationsVol. 4, No. 4, pp.345-360 , 1993.
[2] W. Mao and M. D. Ciletti, "Reducing correlation to improve coverage of delay faults in scan-path design,"IEEE Trans. on Computer-Aided Design ofIntegrated Circuits and Systems, vol. 13, no. 5, pp. 638-646, May 1994
[3] K. Cheng, "Transition Fault Testing for Sequential Circuits," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 12, no. 12, pp. 1971-1983, Dec. 1993.
[4] R. Wilson, "Delay-Fault Testing Mandatory, Author Claims," EE Design, Dec. 2002.
[5] G. Aldrich and B. Cory, "Improving Test Quality and Reducing Escapes," in Proc. Fabless Forum, Fabless Semiconductor Assoc., pp. 34-35, 2003.
[6] N. Ahmed, C.P. Ravikumar, M. Tehranipoor and J. Plusquellic, "At-Speed Transition Fault Testing With Low Speed Scan Enable," in Proc.IEEE VLSI Test Symp. (VTS'05), pp. 42-47, 2005.
[7] S. Wang and S. T. Chakradhar, "Scalable scan-path test point insertion technique to enhance delay fault coverage for standard scan designs," in Proc.IntTest Conf. (ITC'03), pp. 574-583, 2003.
[8] Synopsys DFT Compiler, "User Manual for SYNOPSYS Toolset Version 2004.06," Synopsys, Inc., 2004.
[9] J. Saxena, K. M. Butler, J. Gatt, R. Raghuraman, S. P. Kumar, S. Basu, D. J.Campbell, J. Berech, "Scan-Based Transition Fault Testing – Implementation and Low Cost Test Challenges," in Proc. International Test Conference (ITC'02), pp. 1120 - 1129, Oct. 2002.