The Journal of Engineering Research Vol. 7 No. 2 (2010) 69-79

Design of a Realistic Test Simulator For a Built-In Self Test Environment

A. Ahmad* and D. Al-Abri

** Department of Electrical and Computer Engineering, College of Engineering, Sultan Qaboos University P. O. Box 33, Postal Code 123; Muscat, Sultanate of Oman*

Received 14 May 2009; accepted 16 March 2010

تصميم محاكي اختبارات واقعي لبيئات الاختبار الذاتي

أ. أحمد* و د. د. العبري

ملخص: تعرض هذه الورقة طريقة واقعية لاختبار التصاميم قابلة الاختبار والبيانات ذات قابلية الاختبار الذاتية والتي توجت ببناء محاكي قادر على تحقيق هدف الاختبار للنظام تحت الاختبار . يعتمد المحاكي نهج تشخيص الخطأ مع نظام تقيم للاخطاء لعمل الاختبار . تم تطوير العمل على حاسب اعتيادي دون الحاجة الى اية برامج خاصة مما يجعل الطريقة اقتصادية القيمة. المحاكي مناسب جدا لايجاد متسلسلة واقعية من الاختبارات لتحقيق هدف الاختبار لاي نظام تحت الاختبار (شصه). المحاكي يشتمل على واجهة رسومية للمستخدم ويمكن استعماله دون الحاجة الى اية مهارات برمجية خاصة . تم اختبار المحاكي باستخدام العديد من الدوائر المعيارية. علاوة على ماسبق يمكن استخدام المحاكي كأداة تعليمية للعديد من المقررارت الدراسية مثل الحوسبة المرنة مع الاخطاء تشخيص الاخطاء الالكترونيات الرقمية وتصاميم المنطق الامنة والموثوقة والقابلة للاختبار.

المفردات المفتاحية: اختبار النظم الرقمية ، الاختبار الذاتي ، التصميم لاجل قابلية الاختبار، متجهات الاختبار، تشخيص الخطأ ، طي الخطأ ، الاختبار الواقعي ، تغطية الخطأ ، التكرار.

Abstract: This paper presents a realistic test approach suitable to Design For Testability (DFT) and Built-In Self Test (BIST) environments. The approach is culminated in the form of a test simulator which is capable of providing a required goal of test for the System Under Test (SUT). The simulator uses the approach of fault diagnostics with fault grading procedure to provide the tests. The tool is developed on a common PC platform and hence no special software is required. Thereby, it is a low cost tool and hence economical. The tool is very much suitable for determining realistic test sequences for a targeted goal of testing for any SUT. The developed tool incorporates a flexible Graphical User Interface (GUI) procedure and can be operated without any special programming skill. The tool is debugged and tested with the results of many bench mark circuits. Further, this developed tool can be utilized for educational purposes for many courses such as fault-tolerant computing, fault diagnosis, digital electronics, and safe - reliable - testable digital logic designs.

Keywords: Digital system testing, Built-in self test, Design for testability, Test vector, Fault diagnosis Fault collapsing, Realistic test, Fault cover, Iteration

*Corresponding author's e-mail: afaq@squ.edu.om

The Journal of Engineering Research Vol. 7 No. 2 (2010) 69-79

Nomenclature

ATE	-	Automatic Test Equipment
BIST	-	Built-In Self Test
CAD	-	Computer Aided Design
CUT	-	Circuit Under Test
DFT	-	Design For Testability
FC	-	Fault Coverage
FM	-	Fault Matrix
GND	-	Ground
IC	-	Integrated Circuit
PI	-	Primary Inputs
PO	-	Primary Outputs
s-a-0	-	Stuck at 0
s-a-1	-	Stuck at 1
SOC	-	System On Chip
SUT	-	System Under Test
GUI	-	Graphical User Interface
LSI	-	Large Scale Integration
VLSI	-	Very Large Scale Integration

1. Introduction

Over the years there has been a remarkable growth in Very Large Scale Integration (VLSI) techniques. Tremendous advances in fabrication technologies have led to increased Integrated Circuit (IC) densities. The overall impact has been the decrease in cost per function of digital processing hardware and greater system reliability (Ahmad, 2005). However, as a consequence of higher integration densities, circuits have become very complex. To verify their correct functioning, ICs are required to be thoroughly tested. The cost involved in testing of even Large Scale Integration (LSI) chips is a substantial portion of the total manufacturing cost (Ahmad, 1989). Further, advances in IC technology are occurring at a rate faster than those in test technology. A direct consequence of this is that testing methods, which are inadequate even for LSI circuits, are grossly unable to cope with the increased circuit sizes because of higher chip densities (Abdi, *et al*. 2008; Ahmad, 1989 and Ahmad, 2005).

Formal Design-For-Testability (DFT) techniques are concerned with providing access points for testing (*ie*. enhancing controllability and observability in ICs) (Al-Lawati *et al*. 2004 and Ahmad, *et al*. 2004). Currently, DFT and BIST have become effective and widely acceptable tools for tackling test problems for VLSI chips and systems (Zorian *et al*. 2000). This is because of the inherent advantages of DFT and BIST techniques (Zorian *et al*. 2000). However, any technique chosen must be incorporated within the framework of a powerful Computer Aided Designed (CAD) system providing semi-automatic analysis and feed back, such that the Rule of Ten can be kept under control: if one does not find a failure at a particular stage, then detection at the next stage will cost 10 times as much! (Ahmad 2005).

Testability is a measure to determine a desired degree of accuracy with which the functionality of any system or circuit or component can operate. A very high cost of test, and an uncertain level of delivered system quality are the indirect costs of non DFT design testability. Further, adding the time spent trying to diagnose the fault makes a non DFT design very expensive. When testability is introduced at the design stage, it dramatically lowers the cost of test and the time spent at test. Properly managed, testability enhances the assurance of product quality and smoothes production scheduling assuring high availability of service, and less maintenance cost.

Since testability is not a technological innovation, the preparedness of a mindset that motivates and creates a constant awareness of the importance of ease-of-testing at all levels of developments and use of the systems is essential. In real sense it can be said that testability is a very critical parameter to the manufacturing process of any system and a system that cannot be readily tested is not really to be considered as manufacturable engineering.

Further elongating the problem of testing is at present, in general most system designers and electronics engineers have little knowledge about testing, and thus the companies frequently hire test technology experts to guide their designers on test problems and the com-

The Journal of Engineering Research Vol. 7 No. 2 (2010) 69-79

panies are even bound to pay higher remunerations to the test experts than to the total gross salaries of their designers.

This reflects also today's university education and their designed curriculum. Due to this weakness of designed curriculum, everyone learns about design, but only truly dedicated students learn about test. The next generation of engineers involved with design technology should be made aware of the importance of test and trained in test technology to enable them to produce high quality and defect-free products.

This paper is an alternate effort for presenting an orientation of a conceptual teaching methodology to bridge the gap between the designers and test engineers. Many researchers have put effort to this direction (Ahmad and Al-Abri, 2005; Ahmad et al. 2006 and Ali *et al*. 2005), (Blyzniuk *et al*. 1998; Dariusz, 1998 and Devadze *et al*. 2002), Ivask *et al*. 1998; Jutman *et al*. 2002; Liu *et al*. 2008; Su *et al*. 2008; Ubar 1998; Ubar *et al*. 2002; Ubar and Orasson, 2003 and Ubar and Wuttke, 2001). Since Automatic Test Equipments (ATEs) are costly and not affordable to many organizations, as viable option the process of simulation is usually adapted. This gap in the market has called for this prototype testers and simulated tools (Ahmad and Al-Abri, 2009; Ahmad *et al*. 2006 and Ali *et al*. 2005), (Liu *et al*. 2008 and Su *et al*. 2008). However, such developed tools cannot cope with exhaustive tests because the application of 2^n test vectors to a device with n inputs is not effective if n is large. As the number of tests, 2^n grows exponentially with n, the number of tests required increases rapidly. For n inputs the truth table contains 2^n rows. If a test application is at 1 test/μs, this would take 18 min for n = 30, 13 days for n = 40, and 36 years for n = 50. Thus it is quite obvious that the method of exhaustive testing is unacceptable for testing. We developed a test tool which applies an alternative approach based on the observation that in a SUT, any particular input test vector will usually cover a significant number of faults (Ahmad and Al-Abri, 2009). Any random selection of tests of reasonable size therefore can be expected to achieve reasonable fault coverage. Hence this paper advocating for creating specific tests for faults most likely to occur. Also, none of the researched test simulator has been used the most commonly available platform of EXCEL. We chose EXCEL due to portability, economy, wide storage space and its easy interface with other programming languages like C++, MATLAB, Visual Basic and Java. We have already demonstrated its WebCT applicability through our research work (Ahmad *et al*. 2006). This paper is further extension to our work by incorporating test pattern generation and fault simulation algorithms in cost effective manner. The features included in this tool are described in Section 3 of this paper.

Hence, our developed test tool through this paper is very much helpful for both the electronic embedded industries along with the educational institutions. The presented paper considers how to improve the skills of students to be educated for hardware, embedded and System On Chip (SOC) design in test related topics. Although, here in this paper, our discussion will be restricted on only one component of the tool but the tool has multifaceted usage in different courses with different usages like digital system simulation, fault insertion, test sequence generation, fault matrix generation, fault coverage, fault diagnostic tree generation, predicting targeted goal of test, to find hard faults, fault collapsing sets, fault equivalence sets, test generation and many more topics of interest in various subjects.

Besides the section of introduction, the structure of the paper consists of terminologies, basic concepts of testing and assumptions in Section 2. Section 3 is devoted towards the development of the test tool, its functionality and software architecture of the tool. Simulation experiment and analysis are embodied in Sections 4 and 5 respectively. The conclusions and future research highlights are presented in the Section 6.

2. Basic Concepts And Acquitance Of Terminologies

Definition 1:
Fault is defined as a physical failure mechanism due to some defects of the circuit.

Definition 2:
Error is defined as the condition (or state) of a system containing a fault (*ie*. that the system deviation from the correct state).

In general, the physical defects are modeled by using faults. In particular, logic-level models, such as *stuck-at*, are adapted to measure circuit testability. Functional verification based on simulation aims at detecting faults of circuits.

Definition 3:
Circuit simulation is a process of defining the digital circuit into its primitive gate level structure.

Definition 4:
Fault insertion is a process where every possible fault is simulated into the SUT. A simulated fault is said to be injected into the circuit of the system.

Definition 5:
Fault detection is a process of comparing the response of a known good Version of the circuit to that of the actual circuit existed in an environment of fault,

The Journal of Engineering Research Vol. 7 No. 2 (2010) 69-79

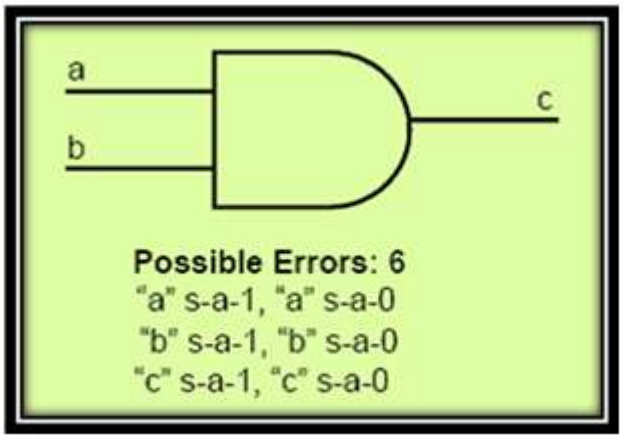

Figure 1. 2 inputs AND gate

FM =

TEST ab	FF	a s-a-0	a s-a-1	b s-a-0	b s-a-1	c s-a-0	c s-a-1
T1 = 00	0	0	0	0	0	0	1
T2 = 01	0	0	1	0	0	0	1
T3 = 10	0	0	0	0	1	0	1
T4 = 11	1	0	1	0	1	0	1

(2)

for a given stimulus set (TEST). A fault is detectable by the TEST if there exist any difference between fault free and faulty response. The process is repeated for each stimulus set and the Fault Matrix (FM) can be generated.

Definition 6:

The Stuck-At (s-a) Fault Model is used in functional testing. This fault model uses the single s-a model, the most common fault model used in fault simulation because of its effectiveness in finding many common types defects. Stuck-At fault model behavior occurs if the terminals of a gate are stuck at either a high [logic 1 or Vcc voltage] or a low [logic 0 or GND voltage]. If it is s-a at one, it is denoted by s-a-1 whereas, if it is s-a zero, it is denoted by s-a-0. The fault sites for this fault model include the pins of primitive instances.

Example 1:

Figure 1 shows the possible s-a faults that could occur on a single AND gate. There are total 6 faults. The 'a' and 'b' are known as Primary Inputs (PI), whereas, 'c' is known as Primary Outputs (PO). Since PI = 2, PO = 1 thus, the size of FM = 4x7.

In general, the size of FM can be defined as in Eq. (1) below.

$$\text{Size of } FM = 2^n * (2 * L + 1) * m \quad (1)$$

Where n is the number of PI, m is the number of PO and L is the total connecting lines in the SUT. The FM for Fig. 1 can be given as:

Definition 7:

Fault Coverage (FC) is defined as the percentage of faults detected from among all faults that the test pattern set tests.

The Table 1 below shows the FC capability of each TEST.

Table 1. FC capability

TEST	FC (%)	Detected faults
T1	16.67	c/ s-a-1
T2	33.33	a/ s-a-1, c/ s-a-1
T3	33.33	b/ s-a-1, c/ s-a-1
T4	50.00	a/ s-a-0, b/ s-a-0, c/ s-a-0

It is also to be observed that the different faults can be tested via different TESTs. Table 2 shows that how faults are covered by different TESTs.

Table 2. Faults detected by different TESTs

Faults	TEST
a/ s-a-0	T4
a/ s-a-1	T2
b/ s-a-0	T4
b/ s-a-1	T3
c/ s-a-0	T4
c/ s-a-1	T1, T2, T3

The Journal of Engineering Research Vol. 7 No. 2 (2010) 69-79

Definition 8:
An un-testable or redundant fault is a fault for which no TEST can exist. Usually, un-testable faults cannot cause functional failures, so the testing tools exclude them when calculating test coverage.

Example 2:

As an example consider the SUT of Fig. 2. In this circuit, signal G always has the value of 1, no matter what the values of A, B, and C. If D is s-a-1, this fault is undetectable because the value of G can never change, regardless of the value at D.

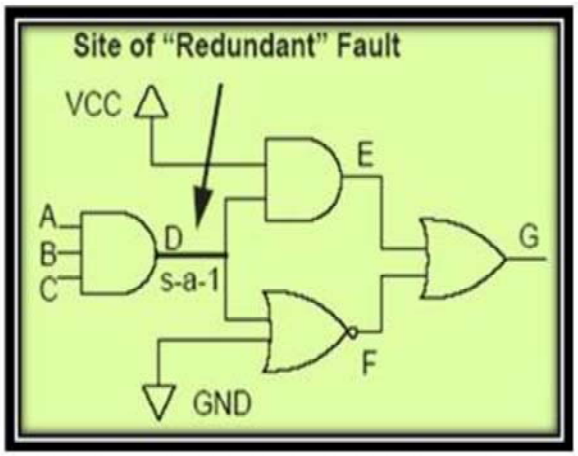

Figure 2. An example of redundant fault

2.1 Assumptions

Some of the assumptions which are considered by the test techniques to test the systems are:

a) Stuck-at fault model is considered in the system.
b) In the systems the gate delays are assumed to be zero.
c) Systems to be tested are limited to combinational circuits only.
d) Un-testable / redundant faults are isolated from fault cover of the system.
e) Binary up counter is used as test sequences to test the system.
f) Single fault model is considered in the system.

3. The Developed Tool

The developed tool named as "Digital Tester1 v7 (2008)" is an updated version of the software introduced through the reported research work (Ahmad and Al-Abri, 2009 and Ahmad *et al*. 2006). The concept of fault equivalence and collapsing is adapted in the development of the tool.

As the concept says that two faults of a SUT circuit are called equivalent if they have exactly the same set of tests and transform the circuit such that the two faulty circuits have identical output functions. As an example consider a two input AND gate with input lines a, b and output c. The stuck-at-0 faults on any input line (line a or line b) of this AND gate can lead to a zero output which is the same faulty result of the stuck-at-zero fault on this gate's output line c, and these three faults have the same test vector (1, 1) therefore, for a two input AND gate, the two input stuck-at-0 and one output stuck-at-0 faults are equivalent. Because of the in-distinguish-ability of equivalent faults, only one of them needs to be tested.

Using the conception of fault equivalence, we can collapse or delete most of equivalent faults. This procedure is called equivalence fault collapsing, which partitions a circuit into disjoined equivalence sets, chooses one fault from each equivalent sets, and forms an equivalent collapsed fault set. For example, because of the equivalence of the three stuck-at-0 faults in case of 2-input AND gate, two of them can be collapsed. These concepts speeds-up the process of testing.

The current tool is now equipped with many more features like:

==> Tracing fault equivalence set,
==> Finding fault collapsing set,
==> Incorporation of a faster fault insertion mechanism,
==> Describing structure models in easy manner,
==> Best search for the targeted test goal, and
==> Realistic TEST search.

3.1 Description Of Basic Functions

Figure 3 shows the first working window which is dedicated to receive general information about the numbers of inputs; structure levels and number of iterations. As soon as entries are complete it will generate the levels and will ask about the number of the gates in each of the levels which can be viewed in the pictorial example (Example 3, Fig. 4).

According to the information entries the structure model can be entered by clicking the "Generate Structure". This activated window "Structure" will only ask the nature of gate logic and the inputs to this gate in prescribed levels of the structure model. It also provides the provision of entering ONLY a one start TEST sequence.

As the description of the SUT structure model is complete, the FM can be generated by clicking the "Generate Fault Matrix". A quick fault insertion procedure is adapted in generating FM. All TESTs or partial TESTs can be opted during the generation of the FM procedure. The window can be visualized by activating "Fault Matrix" from the task bar. The FM also provides about the testability of each TEST along with the information of fault collapsing and equivalence.

The Journal of Engineering Research Vol. 7 No. 2 (2010) 69-79

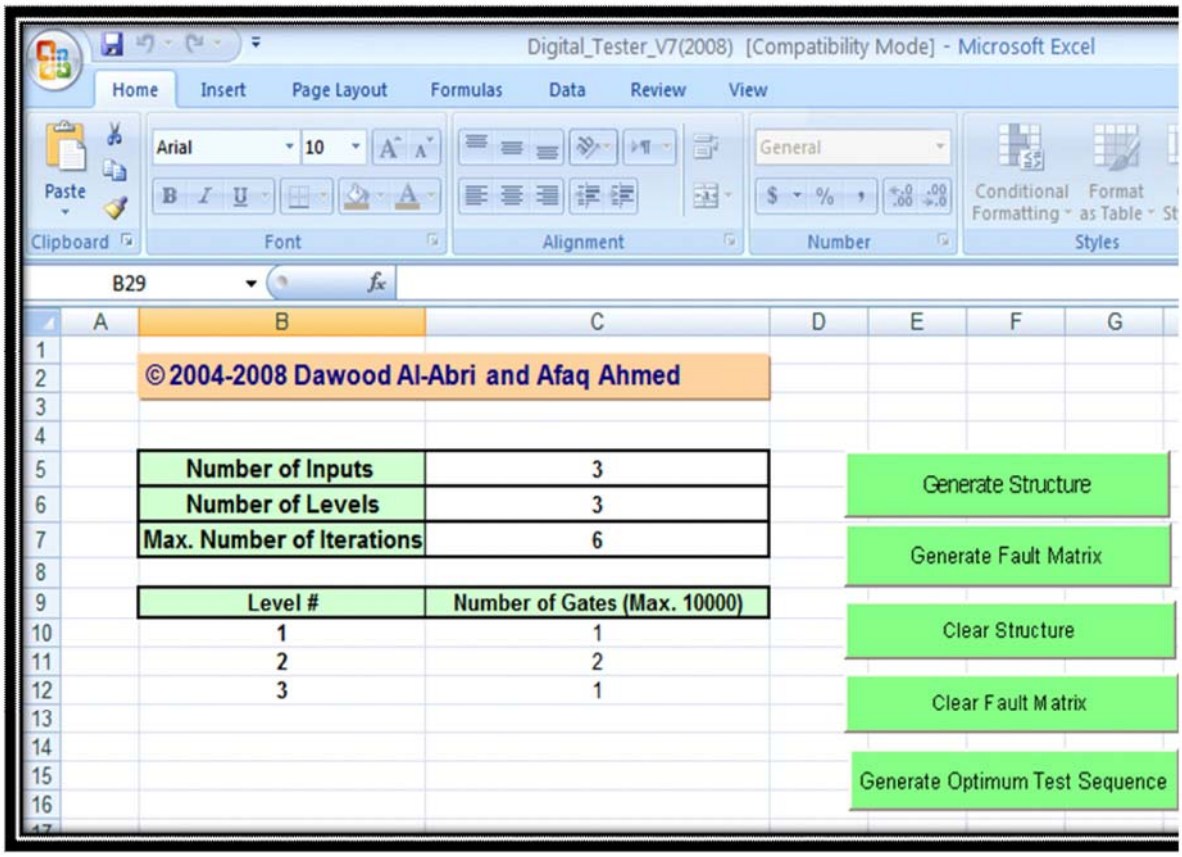

Figure 3a: Working window – Information (Info)

Figure 3b: Task Bar of the window of Figure 3a

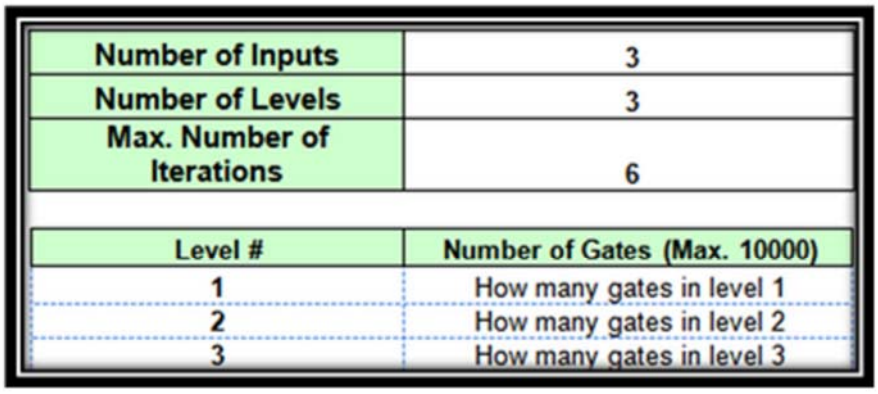

Figure 4. The window created for example 3

The provisions are also provided for clearing the existing structure and the fault matrix. Figure 3a marks the buttons with the names as; "Clear Structure" and "Clear Fault Matrix".

Realistic TEST can be generated by clicking the command button "Generate Test Sequence". This TEST is linked with the maximum test iteration procedure restriction. The complete iteration outputs can be

The Journal of Engineering Research Vol. 7 No. 2 (2010) 69-79

viewed by activating "Iterations" from the task bar. Finally, the testability results can be outputted by clicking the "Opt Testing" from the task bar.

Example 3

A 3-input and 3-level structure of digital system is considered with targeted goal of maximum 6 iterations.

The overall architecture of the developed software tool is described in the ensuing section (Section 3.2).

3.2 The Architecture of Developed Software Model

The architecture of the software is based upon two basic principles; user input and backend computation. Each user input triggers the actions like generation of circuit structure and generation of fault matrix. Figure 5 depicts these actions and user interface data.

The portion of architectural design from fault matrix onwards is shown in Fig. 6 where the programmed algorithmic test generation is adapted to generate each iteration output. Based on user requirement the iteration output presents the statistical summary of the test. Each of the steps requires programs based on knowledge of system simulation, fault insertion, incorporation of test generation algorithms, fault collapsing, fault equivalence, optimization and iteration process. A sample of the backend computation for fault matrix generation is briefly given in Fig. 7.

4. Simulation Experiments

There are many standard test bench circuits of ISCAS85 (Hansen et al. 1999) are simulated using this developed software. It is complex to present the logical diagram of those circuits. Here we present below a case study to make it more readable and beneficial for researchers for the usefulness of the tool for their research works.

4.1 Simulation Run

To demonstrate the simulation run, a circuit shown in Fig. 8 is considered which has 3 inputs, 1 output, 3 levels of model structure, 4 logic gates, and 16 possible faults. After defining the structure model (via "Generate Structure"), a FM is generated (via "Generate fault Matrix") which is shown in Fig. 9. By activating the "Generate Test Sequence" with required goal of maximum six iterations, the result is outputted as shown in Fig. 10. The results of each of the iterations can also be seen just by clicking iterations from the task bar.

5. Analysis of Results

Table 3 shows the maximum possible test vectors {Ti} for i = 1 to 8 with respect to primary inputs of circuit shown in Fig. 8. Figure 11 demonstrate the fault detect-ability (fault coverage) of each tests obtained via the fault matrix result shown in Fig. 9.

As it can be seen from the results of Fig. 10 that there requires only 3 tests to achieve the 100% of the fault coverage. The first TEST selected T7 {x3=1, x2=1, x1=0} which provides 43.75% of the FC. This is the first iteration on the basis of the maximum element, conflict resolution and the best intersection of the sets. On the same way of search of algorithmic procedure T5 {x3=1, x2=0, x1=0} is selected which enhances the FC to 81.25%. Finally, T2 {x3=0, x2=0, x1=1} is selected to reach the target of 100% FC.

As it can be visualized from Figure 12 that although the test time and cost was allocated for a maximum of six iterations (user's set test cost) however, due to the algorithmic design it ends at 3rd iteration process (optimum test cost) and achieves the target of 100% fault coverage. This means even saving the 50% of the provision allocated for the test. Further, it can be visualized that only 37.25% of test requirement is needed to achieve the goal. Thereby, the designed algorithmic procedure saves the 62.75% of the test cost as well as the test time in comparison to the exhaustive test approach of 100% test cost (maximum test cost).

Conclusions

The developed tool considers all effective points of test procedures, like target, fault collapse, given constraints, fault equivalence, best intersection sets and enumerating the data for each steps and hence very useful. The developed tool is dynamic in nature since it keeps in mind about the pace of the electronic industries. And, therefore, we made a provision of applying 14 inputs; accommodating 63 levels of structured layers of digital systems where as in each of the levels 10000 gates can be defined. The nature of these gates may be of any kinds logic gates such as: NOT, OR, AND, NOR, NAND, Exclusive OR, and Exclusive NOR. The provisions of the latches and flip-flops are also incorporated in the developed tool. The developed tool is user friendly. These powerful attributes of this developed tool and as well its enormous applications makes this an excellent learning tool for the courses in the area of digital system testing in an eLearning environment along with the usage of electronic industries for incorporating the test procedures in the embedded systems. Since the DFT and BIST environments incorporate the test circuitry in its embedded design therefore, this developed tool will be very much cost effective measure in designing the digital circuits in DFT and BIST environments.

We developed the software tool in Excel 2007 which is limited to 2 Gigabytes of memory for the Excel process under Windows XP. This 2 Gigabyte limit is a

The Journal of Engineering Research Vol. 7 No. 2 (2010) 69-79

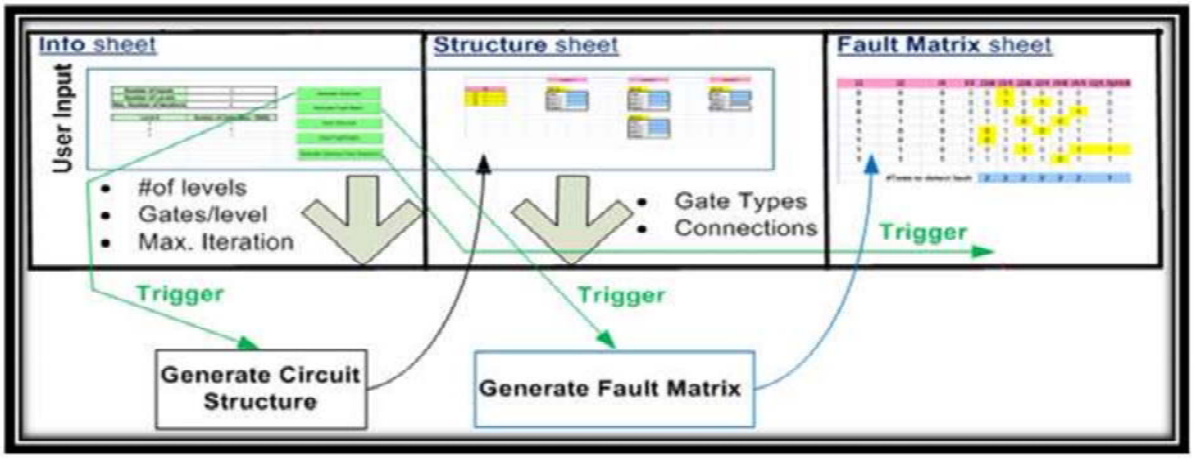

Figure 5. The user input and trigger actions

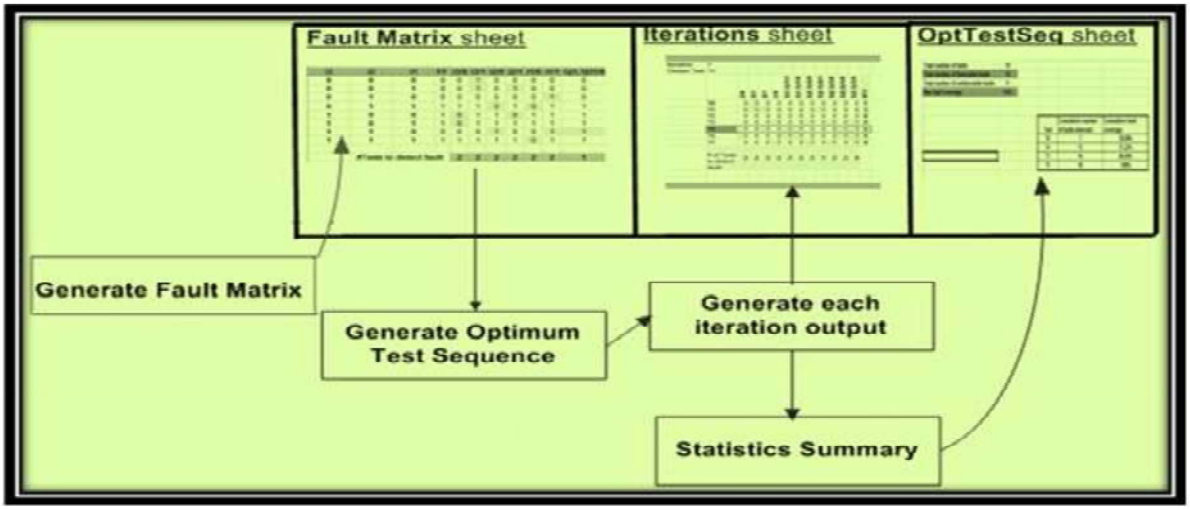

Figure 6. The program outputs

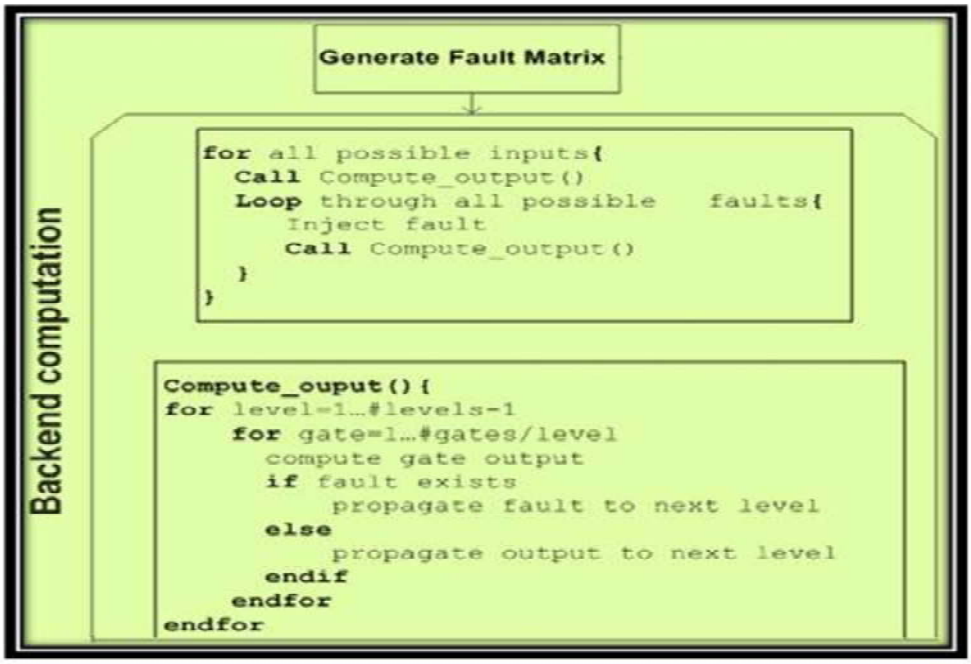

Figure 7. The beckend computation algorithm for fault matrix generation

The Journal of Engineering Research Vol. 7 No. 2 (2010) 69-79

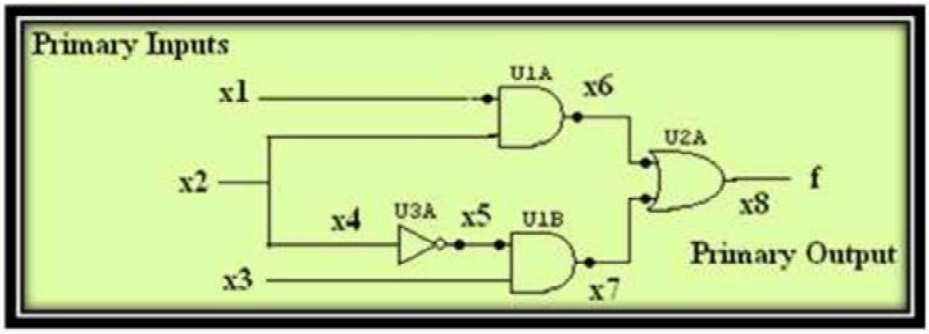

Figure 8. A model for demonstration

x3	x2	x1	FF	x3/0	x3/1	x2/0	x2/1	x1/0	x1/1	x4/0	x4/1	x5/0	x5/1	x6/0	x6/1	x7/0	x7/1	x8/0	x8/1	# FC	Faults Detected
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0	1	4	x3/1,x6/1,x7/1,x8/1
0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	1	0	1	0	1	5	x3/1,x2/1,x6/1,x7/1,x8/1
0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0	1	4	x1/1,x6/1,x7/1,x8/1
0	1	1	1	1	1	0	1	0	1	1	1	1	1	0	1	1	1	0	1	4	x2/0,x1/0,x6/0,x8/0
1	0	0	1	0	1	1	0	1	1	1	0	0	1	1	1	0	1	0	1	6	x3/0,x2/1,x4/1,x5/0,x7/0,x8/0
1	0	1	1	0	1	1	1	1	1	1	0	0	1	1	1	0	1	0	1	5	x3/0,x4/1,x5/0,x7/0,x8/0
1	1	0	0	0	0	1	0	0	1	1	0	0	1	0	1	0	1	0	1	7	x2/0,x1/1,x4/0,x5/1,x5/1,x7/1,x8/1
1	1	1	1	1	1	1	1	0	1	1	1	1	1	0	1	1	1	0	1	3	x1/0,x5/0,x6/0,x8/0
TEST for faults				2	2	2	2	2	2	1	2	2	1	2	4	2	4	4	4		

Figure 9. The FM for the model of Figure 5

Total number of faults	16	
Total number of Detectable faults	16	
Total number of undetectable faults	0	
Max fault coverage	100%	
OPTIMAL TEST SEARCH		
TEST	Cumulative FC	% Cumulative FC
T7	7	43.75%
T5	13	81.25%
T2	16	100.00%

Figure 10. Test results

limit on the Virtual Memory address space. Virtual memory used by a process is larger than the working set memory reported by Windows Task Manager, so the amount of useable memory under Excel 2007 is considerably a little less than twice that of Excel 2003. Increased spreadsheet row and column capacity of one million rows by 16,000 columns enables the user to import and work with massive amounts of data and

The Journal of Engineering Research Vol. 7 No. 2 (2010) 69-79

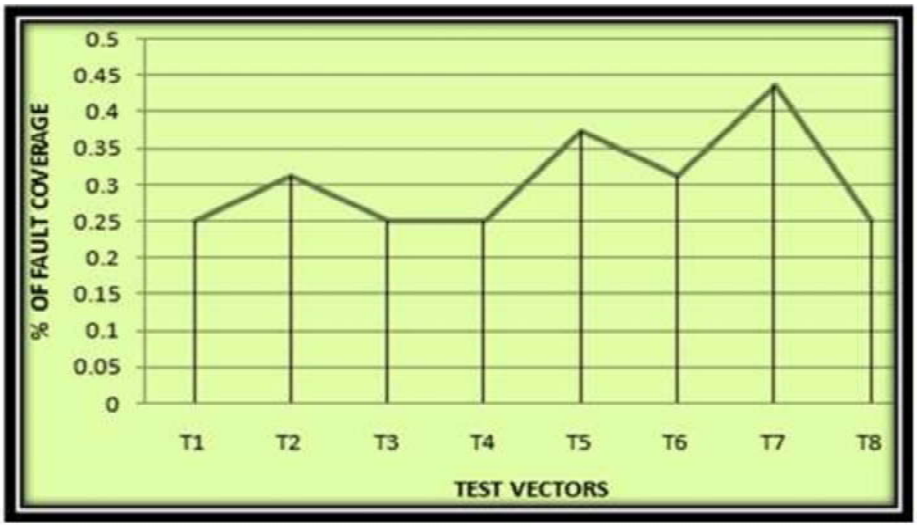

Figure 11. % of fault coverage of each test

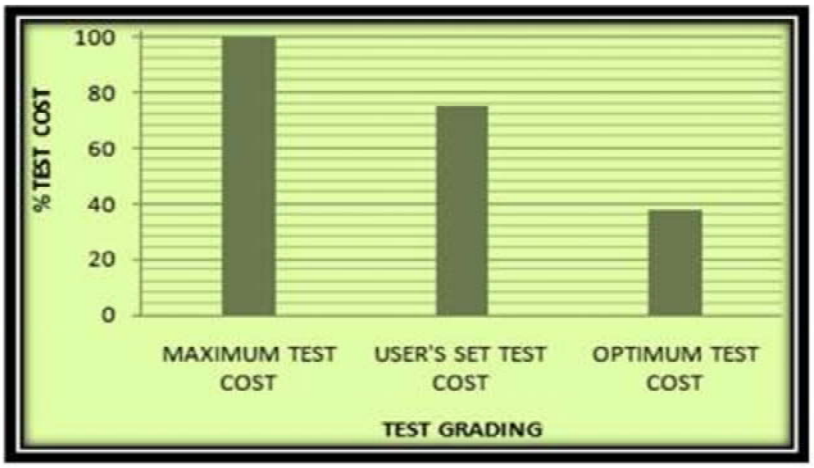

Figure 12. Test grading

Table 3. Possible test vectors

TEST VECTORS	INPUTS		
	x3	x2	x1
T1	0	0	0
T2	0	0	1
T3	0	1	0
T4	0	1	1
T5	1	0	0
T6	1	0	1
T7	1	1	0
T8	1	1	1

achieve faster calculation performance. Therefore, the developed tool can cope with large complex SUT in efficient manner.

Our further orientation of the research work is to extend the capability of the tool to compute the reliability, controllability, observability, availability and maintenance scheduling of digital systems.

Acknowledgments

The acknowledgments are due to Sultan Qaboos University (Sultanate of Oman) for providing research support grant (SQU-DVC/ PSR/RAID/2010/23).

References

Abdi, A., Tahoori M. B. and Emamian, E. S., 2008, "Fault Diagnosis Engineering of Digital Circuits Can Identify Vulnerable Molecules in Complex Cellular Pathways", Sci. Signal., Vol. 1(42), pp 1 - 10.

Ahmad, A., 1989, "On a Design Approach for

Reducing Aliasing Errors and Achieving Higher Testability Goals in Combinational Circuits", Indian Institute of Technology, Roorkee Ph. D. Thesis.

Ahmad, A., 2005, "Testing of Complex Integrated Circuits (ICs) - The Bottlenecks and Solutions", Asian Journal of Information Technology, Vol. 4(9), pp. 816 - 822.

Ahmad, A., Al-Abri, D., 2009, "Design of Dynamic Test Tool in the Area of Digital System Testing", Presented and Published in the Proceedings of International Conference on Computer, Communication and Power (ICCCP'09) held at Oman, pp. 1-4 (Computer and Network Systems).

Ahmad, A., Al-Abri, D. and Al-Ramhi, M. M., 2006, "Design of an E-Learning Process in the Area of Digital System Testing", Presented and Published in the Proceedings of International Conference on Distance Education (ICODE2006), held at Muscat, Sultanate of Oman, March 27 - 31, 2006, pp 1-8.

Ali, L. Sidek, R., Ishak, A. Alauddin, M. A., and Bambang, S. S., 2005, "Design of a Low Cost IC Tester", American Journal of Applied Science, Vo. 2(4), pp. 824 - 827.

Al-Lawati, A. M. J. and Ahmad, A., 2004, "Realization of a Simplified Controllability Computation Procedure - A MATLAB-SIMULINK Based Tool", Sultan Qaboos University Journal for Scientific Research - Science and Technology, Oman, Vol. 8, pp. 131 - 143.

Ahmad, A., Al-Lawati, A. M. J. and Al-Naamany, A. M., 2004, "Identification of Test Point Insertion Location via Comprehensive Knowledge of Digital System's Nodal Controllability Through a Simulated Tool", Asian Journal of Information Technology, Vol. 3(3), pp. 142 - 147.

Blyzniuk, M., Cibakova, T., Gramatova, E., Kuzmicz, W., Lobur, M., Pleskacz, W., Raik, J. and Ubar, R., 2000, "Hierarchical Defect-Oriented Fault Simulation for Digital Circuits", IEEE European Test Workshop, Cascais, Portugal, pp. 151-156.

Dariusz, B., 1998, "How Faults can be Simulated in Self-Testable VLSI Digital Circuits?", Proceedings EUROMICRO'98, Vol. 1, pp.10180.

Devadze, S., Jutman, A., Sudnitsyn, A., Ubar, R., Wuttke, H. D., 2002, "Teaching Digital RT-Level Self-Test Using a Java Applet", 20th IEEE Conference NORCHIP'2002, Copenhagen, Denmark, pp. 322-328.

Hansen, M. C., Yalcin, H. and Hayes, J. P., 1999, "Unveiling the ISCAS-85 Benchmarks: A Case Study in Reverse Engineering", IEEE Design & Test, Vol. 16(3), pp. 72 - 80.

Ivask, E., Raik, J., Ubar, R., 1998, "Comparison of Genetic and Random Techniques for Test Pattern Generation", Proceedings of the 6th Baltic Electronics Conference, Tallinn, pp. 163-16.

Jutman, A., Raik, J., Ubar, R., 2002, "SSBDDs: Advantageous Model and Efficient Algorithms for Digital Circuit Modeling, Simulation & Test", Proceedings of 5th International Workshop on Boolean Problems (IWSBP'02), Freiberg, Germany, pp. 157-166.

Liu, Y., Su, W. and Fan, T., 2008, "The Digital Circuit Fault Diagnosis Interface Design and Realization Based on VXI", Proceedings ISECS International Colloquium on Computing, Communication, Control, and Management 2008 (CCCM '08), Guangzhou, Vol. 2, pp. 215 - 219.

Su, W., Zhang, S., Xue, L., 2008, "Research and Realization of Digital Circuit Fault Probe Location Process", Proceedings International Conference on Intelligent Computation Technology and Automation 2008 (ICICTA'08), Hunan, Vo. 2, pp. 897 - 900.

Ubar, R., 1998, "Dynamic Analysis of Digital Circuits with Multi-Valued Simulation", Microelectronics Journal, Elsevier Science Ltd., Vol. 29(11), pp. 821-826.

Ubar, R., Jutman, A., Orasson, E., Raik, J., Wuttke, H.D., 2002, "Internet-Based Software for Teaching Test of Digital Circuits", Microelectronics Education, Marcombo U Boixareu Ed., pp. 317 - 320.

Ubar, R., Orasson, E., 2003, "E-Learning Tool and Exercises for Teaching Digital Test", Proceedings of 2nd IEEE Conference on Signals, Systems, Decision and Information Technology, Sousse, Tunisia, Vol. CIT-6, pp.1 - 6.

Ubar, R., Wuttke, H.D., 2001, "The DILDIS-Project - Using Applets for More Demonstrative Lectures in Digital Systems Design and Test", Proceedings of 31st ASEE/IEEE Frontiers in Education Conference, Reno, NV, USA, Abstracts, p. 83.

Zorian, Y., Mourad, S., 2000, "Principles of Testing Electronic Systems", J. Wiley & Sons, Inc. New York, 2000, p. 420.