International Journal of Electrical, Electronics and Data Communication, ISSN(p): 2320-2084, ISSN(e): 2321-2950
Volume-6, Issue-7, Jul.-2018, http://iraj.in

TESTING OF STUCK AT FAULT IN DIGITAL CIRCUITS AT REGISTER TRANSFER LOGIC (RTL)

[1]NAINA A. UDGE, [2]S. A. LADHAKE, [3]P. D. GAWANDE

[1]Principal, SIPNA COET, Amravati (M.H.)
[2,3]Associate Professor, SIPNA COET, Amravati (M.H.)

Abstract - Semiconductor Integrated Circuits (ICs) can have millions of digital circuits which can translate to billions of transistors. Lots of effort has been put into Electronic Design Automation (EDA) systems and Design for Test (DFT) techniques to manage the development and application of digital circuit testing. In the beginning these software programs and DFT techniques used the Stuck at Fault Model.With a stuck at fault model we are applying a structural test approach. Instead of testing all combination of 1's and 0's to a VLSI device, we will test with a reduced set of test vectors. Stuck at Fault Models operate at the logic model of digital circuits. We will detect the stuck-at fault using the concept of textio.

Keywords - Stuck-atfaults; Fault coverage; Testpoints; Validation Sets; Textio

I. INTRODUCTION

Being able to design a workable system solution for a given problem is only half the battle unfortunately. We must also be able to test the system to a degree which ensures that we can have a high confidence level that it is fully functional. This is generally not a straightforward task, in very small scale digital systems, we can test exhaustively, that is to say, we can exercise the system over its full range of operating conditions. In a larger scale system, it is no longer possible to do this and therefore we must look at other strategies to ensure that the system will be properly tested. When testing a digital logic device, we apply a stimulus to the inputs of the device and check its response to establish that it is performing correctly. The input stimulus is referred to as a test pattern. In general, we observe the response of the device at its normal output pins, however, it may be that the device is specially configured during the test, to permit us to observe some internal nodes which generally would not be accessible to the user. The response of the device is evaluated by comparing it to an expected response which may be generated by measuring the response of a known good device, or by simulation on the computer. If the device under test (DUT) passes the test, we cannot say categorically that it is a ``good'' device. The only conclusion that we can draw from the device passing a test, is that the device does not contain any of the faults for which it was tested. It is important to grasp this point, a device may contain a huge number of potential faults, some of which may even mask each other under specified operating conditions. The designer can only be sure that the device is 100% good if it has been 100%tested, this is rarely possible in real life systems.

In VLSI circuits, we have a high ratio of logic gates to pins on the device, there is generally no way of accessing most of the logic, so we cannot directly probe the internals of the device. Because of this problem, we need a way of generating tests which, when applied to the inputs of a circuit, give a set of signals which indicate whether or not the device is good or faulty. The set of stimulus input and expected output pattern is called a ``Test Vector''. The test vectors distinguish between the good machine and the faulted machine.

II. BASIC CONCEPTS OF FAULT DETECTION

Fault detection in a logic circuit is carried out by applying a sequence of tests and observing the resulting outputs. A test is an input combination that specifies the expected response that a fault-free circuit should produce[2]. If the observed response is different from the expected response, a fault is present in the circuit. The aim of testing at the gate level is to verify that each logic gate in the circuit is functioning properly and the interconnections are good. Henceforth, we will deal with stuck-at faults only unless mentioned otherwise. If only a single stuck-at fault is assumed to be present in the circuit under test, then the problem is to construct a test set that will detect the fault by utilizing only the inputs and the outputs of the circuit. A test detects a fault in a circuit if and only if the output produced by the circuit in the presence of the fault is different from the observed output when the fault is not present. To illustrate, let us assume that input a of the NAND gate. The output responses of the gate to all input combinations for both fault-free and fault-present conditions are shown in Table 1. It can be seen in Table 1 that only for input combination ab=0, the

Inputs		Outputs	
a	b	c(fault free)	c(fault present)
0	0	1	1
0	1	1	0
1	0	1	1
1	1	0	0

Table 1: Output response of the NAND gate

output is different in the presence of the fault a s-a-1 and when the gate is fault-free. In order to detect a fault in a circuit, the fault must first be excited; that is, a certain input combination must be applied to the circuit so that the logic value appearing at the fault location is opposite to the fault value. Next, the fault must be sensitized; that is, the effect of the fault is propagated through the circuit to an observable output.

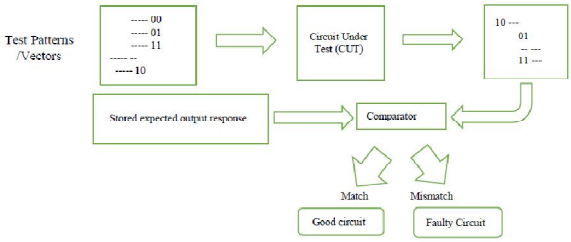

Figure 1

III. PROCEDURES USED BEFORE FOR TESTING

Different methods are used for testing stuck at fault in combinational circuits.

3.1 Adding buffer to each port of RTL circuits

- Firstly test bench is developed and the simulation is done on a good circuit and then on each of the faulty circuits using any simulator[5].
- The outputs obtained in each case of the faulty circuits are compared with the output of the good circuits to determine where fault actually occurred . The fault list is tabulated.
- The ratio of the numbers of RTL faults detected to the total number of RTL faults gives the RTL fault coverage.

3.2 Implementation of Validation Test sets

1. The scheme first derives the controller behaviors from validation test sequences and reuses them for simplifying justification/propagation analysis corresponding to precomputed test vectors/responses of datapath RTL modules.
2. A heuristic is used to identify controller behaviors that are compatible with a given set of precomputed test vectors/responses. It requires only a single pass through the CDFG[6] corresponding to a validation test sequence and is accurate, resulting in a small number of test generation runs.
3. Test generation is performed at the RTL and the controller behavior is prespecified, which results in very small test generation times. First step is identification of compatible controller behaviors consisting of Augmented Fault Simulation to Derive Activation-Detection Time Frame Pair and Analysis of Requirements to Identify Compatible Faults. Next step is SAT-based RTL ATPG is used to obtain a test sequence that reuses the controller behavior to justify and propagate the precomputed test vector and response to primary inputs and outputs, respectively.Testing is the process to verify that the fabricated VLSI circuit functions as expected or designed. Figure1. shows the principle of typical digital testing.

Circuit/Device under test (CUT/DUT) is exercised by applying test signals to its inputs. Output responses are observed and compared. If there's no mismatch between observed response and stored correct response, the CUT is considered good, otherwise it will be labelled as bad. The input data to the CUT is referred to as the test pattern or test vector. Usually tests applied during testing are for fault detection. The purpose of a detection test is to identify the CUT as faulty of fault-free. But in diagnosis, additional test patterns maybe applied, aiming to find the cause/location of the failure. Currently the available commercial silicon fabrication technology reduces the feature size down to 22nm[7]. Process variation has greater and greater impact on product quality. A manufactured circuit may develop various kinds of defects during and after the fabrication process.

3.3 Use of Scan Design

A commonly used testing strategy is to structurally test every signal line (input/output of logic gate), to see whether they toggle correctly as simulated. To do this we need two prerequisites: controllability and observability. The former indicates that a test pattern/sequence should be able to set the targeted line to a desired value. And the latter means we should be able to observe the behavior of a line from a primary output or an observation point such as a scan flip-flop.With scan design we can shift in any test pattern from a scan-in pin and observe the output values captured in flip-flops from a scan-out pin. The area overhead is not significant. And only several additional pins are needed (scan-in, scan-out, scan-enable, etc.), which are very valuable resources in digital design. State-of-the-art VLSI chips have billions of transistors and thousands of input/output pins. It is very difficult to add dedicated pins just for testing. Quite often, even with scan chains implemented, some part of the circuit is still uncontrollable/unobservable. Ad-hoc techniques are applied in these situations to increase the fault coverage to nearly 100%. Nowadays people put much emphasis on reliability of electronic devices, thus near 100% fault coverage, along with low DPPM (defective parts per million), is usually required. To achieve this, control and/or observation points are inserted into the circuit which do not influence the functionality but make it easier to test.

3.4 Built-in Self-Test

Another popular DFT technique is Built-in Self-test (BIST). Test patterns are generated internally inside

the CUT and output responses are compacted into a pass-fail bit or 14 signatures. This means a chip can test itself without outside stimulus, thus greatly reducing the cost of testing, and ATEs are no longer needed. In reality any single DFT technique is not sufficient to achieve near 100% coverage while maintaining a reasonable cost. Most designs implement a combination of several DFT techniques. For example BIST is used for initial screening and then ATE is used to apply additional deterministic test patterns for further testing.

3.5 Automatic Test Pattern Generation

Automatic Test Pattern Generation is another key area in VLSI testing. Its primary role is to generate test patterns with as much coverage as possible. An ATPG usually comes with a fault simulator whose goal is to simulate generated test patterns to find undetected faults, calculate coverage[3], etc. Any ATPG system will benefit a lot from a fast/efficient fault simulator. The goal of ATPG-oriented approaches is to reduce the test generation complexity by increasing the transparency of modules, breaking loops and increasing the controllability/observability of registers.

There has been numerous attempts to generate test vectors at the RTL targeting logic-level stuck at faults as well as various RTL modeled faults .Hierarchical functional HDL circuits have been targeted for test generation. However, the algorithm explicitly targets stuck-at faults at the logic level and there is no explicit RTL fault model used. Each RTL module is tested with a precomputed stuck-at test set from the primary inputs of the RTL circuit. Note that testing of RTL modules using precomputed test sets was first introduced. Similar philosophy is used but the test path generation algorithm makes use of regular expression based analysis. The above RTL techniques require that the precomputed test sets of RTL modules must be stored in a test set library for every bit-width and every implementation of a module possible. Since it is almost impossible to predict a logic implementation of a module in a constraint-driven logic synthesis scenario where the module implementations are essentially technology-mapped boolean equations, theabove methods can be approximate.

Table 1: Benchmark Characteristics

CKT	#Lin.	#Proc.	Type	# Gates	# FFs
B01	110	1	flat	47	5
B02	70	1	flat	29	4
B08	89	1	flat	168	21
B14	509	1	flat	4776	245
Paulin	130	1	flat	39558	227
GPIO	1002	20	hierarchical	1720	148
ATMS	3214	84	hierarchical	8160	1490
MEMX	10674	651	hierarchical	16871	1954

Table 2: Performance (manufacturing test and validation coverages) of logic-level stuck-at test sets at the RTL for area-optimized circuits

CKT	FltC (%)	StmC (%)	BrC (%)	CondC (%)	TogC (%)	OCCOM (%)
B01	100.0	100.0	100.0	100.0	100.0	100.0
B02	100.0	100.0	100.0	100.0	100.0	100.0
B08	99.5	100.0	100.0	100.0	100.0	97.9
B14	95.1	100.0	100.0	99.1	100.0	92.5
Paulin	99.7	100.0	100.0	100.0	100.0	100.0
GPIO	99.4	100.0	100.0	100.0	100.0	100.0
ATMS	95.4	100.0	100.0	99.1	100.0	93.8
MEMX	95.0	100.0	99.8	98.9	100.0	92.1

Other ATPG techniques such as define new fault models at the RTL and generate tests according to that model. They then validate that coverage by doing logic-level fault simulation and show close correlation to the RTL model. In both these cases, a bit-error model is assumed in which each bit in each variable at the RTL is injected with a stuck-at 0 or stuck-at 1 fault. In addition to that, a stuck-at true/false fault is assumed for each condition at the RTL[8]. However, since the focus of the work is exclusively on ATPG, no attempts are made to correlate ATPG and validation coverage metrics.. In this work,a new coverage metric called validation vector grade (VVG) at the RTL is defined and shows that this coverage metric closely tracks fault-coverage at the logic level. The VVG metric is actually a variation of the bit-error model discussed above. In addition to bit stuck-at faults for all variables, it also injects faults at all fanout branches much like logic-level fault simulation.

IV. FAULT SIMULATION

The operation of DUT is analysised on the basis of inputs and outputs. The variation in the output indicates the presence of fault in the circuit. The steps of testing are demonstrated in the below flow diagram:

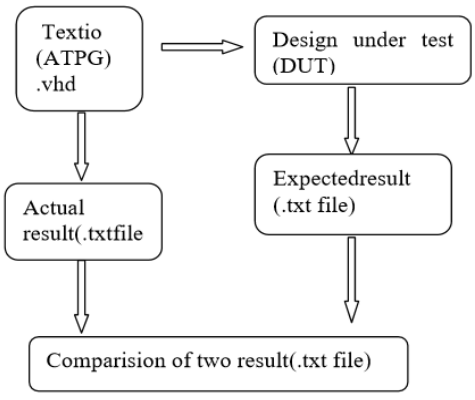

Figure 2

The faulty circuit is simulated first using software modelsim. After this, detection of fault is done by analysis the individual paths of transmission of output inside the circuit, they are known as test points. In our circuit four testpoints are there. Finally, the results of

good circuit i.e. expected results and the results of faulty circuit i.e. actual results , both are compared by simulating the circuit in which expected results are taken as inputs of the circuit.

The values of signals sf_test1, sf_test2, sf_test3, sf_test4 indicates the presence of stuck-at faults in the circuit. The signals values ie. sf_test1, sf_test2, sf_test3, sf_test4, for faulty circuit has the values of testpoints, which indicates the fault:

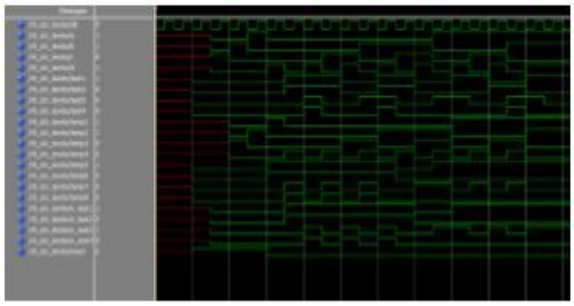

Figure 3

V. RESULT

Here, we are using the concept of textio for the detection of stuck-at fault in the circuit." Textio" is one of the easier way of fault modelling and fault coverage is calculated without any consciousness. Below given output file which is generated after comparison of actual results and expected results indicates the presence of stuck-at fault. In the above table of output, A,B,C,D& RESET are taken as inputs.Test1,Test2,Test3,Test4 are taken as the test points of the circuit. The values of outputs i.e. SF_Test1, SF_Test2, SF_Test3, SF_Test4 indicates the presences of stuck-at fault in the circuit.If there is the fault for any one of the inputs then, SF_Test1,SF_Test2,SF_Test3,SF_Test4 takes the values of Test1,Test2,Test3,Test4 otherwise it has the null 'z' value. Since for the set of 15 inputs, we get the proper outputs and fault is detected for any particular combination of inputs,therefore fault coverage for given circuit is 100%.

reset	A	B	C	D	O_test1	O_test2	O_test3	O_test4	test1	test2	test3	test4	SF_test1	SF_test2	SF_test3	SF_test4
0	1	0	0	0	0	0	0	0	0	0	0	0	z	z	z	z
0	0	0	0	1	1	1	0	0	1	1	0	0	z	z	z	z
0	0	0	0	0	1	1	1	1	1	1	1	1	z	z	z	z
0	0	0	0	1	1	1	0	0	1	1	0	0	z	z	z	z
0	0	0	1	0	1	1	1	1	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	0	0	0	0	1	0	0	0	1	z
0	0	1	0	0	1	1	1	1	1	1	1	1	z	z	z	z
0	0	1	0	1	1	1	0	0	1	1	0	0	z	z	z	z
0	0	1	1	0	0	0	0	0	0	0	0	0	z	z	z	z
0	0	1	1	1	0	0	1	0	0	0	1	0	z	z	z	z
0	1	0	0	0	1	0	1	0	1	0	1	0	z	z	z	z
0	1	0	0	1	1	0	0	0	1	0	0	0	z	z	z	z
0	1	0	1	0	1	0	1	0	0	1	0	0	0	1	0	z
0	1	0	1	1	1	0	0	0	0	1	1	1	0	1	1	1

Table 3: Result of simulation

CONCLUSION

The very first method to fault modelling methodology has used a validation test sets to generate test sequence that have good stuck-at fault coverage for RTL circuits.This method results in very small test generation times. After this method,RTL-ATPG algorithm was presented to generate test vectors for single clock functional RTL design.Thealgorithm uses a data structure called ADD. But here in this paper concept of textio is used to detect the stuck-at fault in RTL circuit.Also, this method results in easier estimation of fault coverage.This leads to the better efficiency of this method than previously used methods of fault coverage in RTL circuits.

REFERENCES

[1] R. C. Ho and M. A. Horowitz, "Validation coverageanalysis for complex digital designs," in Proc. Int. Conf. Comput.-Aided Des., Nov. 1996.
[2] Ghosh, A. Raghunathan, and N. K. Jha, "A design for testability technique of RTL circuits using control/data flow extraction," in Proc. Int. Conf. Comput.-Aided Des., Nov. 1996.
[3] D. J. Moundanos, J. A. Abraham, and Y. V. Hoskote, "Abstraction techniques for validation coverage analysis and test generation.
[4] Indradeep Ghosh and Srivaths Ravi, "On Automatic Generation of RTL Validation Test Benches Using Circuit Testing Techniques" Fujitsu Laboratories of America, Sunnyvale, CA 94086,NECLaboratoriesAmerica, Princeton, NJ 08540, 2001.
[5] Indradeep Ghosh and Masahiro Fujita, "Automatic Test Pattern Generation for Functional Register-Transfer Level Circuits Using Assignment Decision Diagrams" ieee transactions on computer-aided design of integrated circuits and systems, vol. 20, no. 3, march 2001.
[6] L. Lingappan and N. K. Jha, "Unsatisfiability based efficient design for testability solution for register-transfer level circuits," in Proc. VLSI Test Symp., May 2005.
[7] Suma M.S. , K.S. Gurumurthy "Fault Coverage for digital circuits at RTL"2011.
[8] Sarvesh Prabhu, Michael S. Hsiao, Loganathan Lingappan and Vijay Gangaram, "A SMT-based Diagnostic Test Generation Method for Combinational Circuits" ieee 30th vlsi test symposium (vts) 2012.

★★★