Implementation of Combinational Automatic Test Pattern Generator D_Algorithm

Vivek A. Hadge
M.Tech, Dept. Electronics Engineering
R. C. O. E. M., Nagpur

Smita G. Daware
Assistant prof. Dept. Electronics Engineering
R. C. O. E. M., Nagpur

ABSTRACT

Testing of combinational circuit is crucial important to ensure high level of functionality. As density of digital circuit increases rapidly day by day these increases cost and time to test a particular combinational circuit for testing such circuit we need high quality test vector pattern with minimum number of input combination. In this work, we are designing Automatic test pattern generator (ATPG) D_Algorithm which will generate a minimum number of input pattern to detect fault like stuck-at-0 fault, stuck-at-1 fault, short circuit fault. D_Algorithm has been design by writing practical extraction and report language script to generate VHDL coding which is simulated on Xilinx 9.1.

Keywords

ATPG (Automatic Test Pattern Generator), FPGA (Field Programming Gate Arrays), FATE (FPGA Based Automatic Test Equipment), CUT (Circuit under Test)

1. INTRODUCTION

Density of digital circuits increasing rapidly day by day these increases cost and time to test particular circuit. Testing such complex digital circuits we need quality test vector pattern with minimum number input combination. For generating test pattern multiple number of algorithm used like D-algorithm, PODEM etc. important fact about these algorithm is that circuit should be tested with minimum number of input Combinations of test vector. One of the simplest cost efficient and reconfigurable method called FATE (FPGA based Automatic Test Equipment). FATE allows. FATE allows executing digital circuit's test only using laptops and FPGA board and modified according to CUT (circuit under test).Complete test process consist of ATPG, CUT and comparator. ATPG generate an input test pattern for Circuit under Test and comparator compares output response to stored expected output complete test process shown in Fig [1]. ATPG algorithm injects fault into circuit and then uses different techniques to activate fault into circuit.

The rest of the paper has been organized in the following way, section II presents complete mathematical and theoretical ATPG D_Algorithm with flowchart. Section-III present the D_Algorithm theoretical calculation. Section-IV present the implementation and simulation result and finally conclusion are drawn in section V.

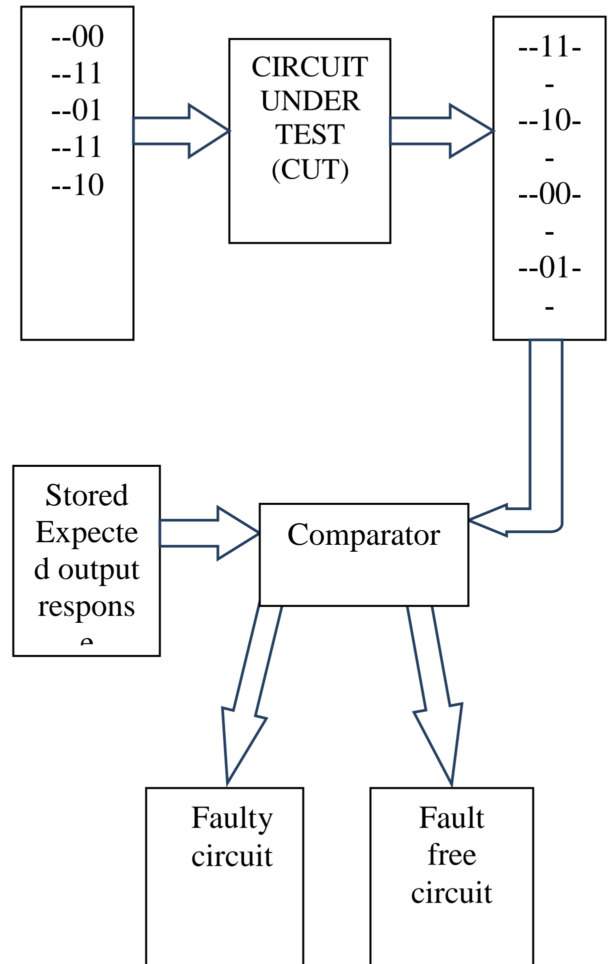

Fig 1: Complete test flow

1.1 ATPG Algorithm

D_Algorithm also known as Roth algorithm [2] an efficient ATPG path sensitization method which consist of Singular cover of logic gate [10] is the minimal set of input signal assignments needed to represent the essential prime implicants in Karnaugh map of that logic gate, for both output case of 0 and 1. Table (1,2) gives singular cover of AND and NOR gate. A D_Cube [10] is collapsed truth table defined as the set of circumstances under which different cube labeling for different logic gate can coexist in the circuit. Here if one cube assign a specific signal value, then other cube must assign either that same signal value or X to entry that can be used to characterize an arbitrary logic block. Combine rows 3 and 1 of the AND gate singular cover, express it in Roth's five valued algebra yield propagation D-cube "D 1 D" by interchanging the role of the two inputs and we get the additional cube as "1 D D" and these two D-cube together we get third D-Cube as "D D D" .Similarly for NOR gate we get d-cube as "D 0 D'" , "0 D D'" and "D D D'." the D intersection operation [9] is the signal. In intersection table Greek symbol ø and ψ represent incompatible assignments and similarly Greek symbols µ and λ indicates incompatibilities if both are present in D-Cubes with multiple input D and D'. Table (3)

represents the D-intersection table. Primitive D-cube [10] of failure model fault in a logic circuit and can model faults as stuck-at-0, stuck-at-1 fault, short circuit fault, Arbitrary change in logic gate function.

Table 1. Singular covers for AND gate

Gate type	Input		Output
AND	A	B	Y
1	0	X	0
2	X	0	0
3	1	1	1

Table 2. Singular covers for NOR gate

Gate type	Input		Output
NOR	C	D	Z
1	1	X	0
2	X	1	0
3	0	0	1

Table 3. D-intersection Table

Intersect	0	1	X	D	D’
0	0	Ø	0	ψ	ψ
1	ø	1	1	ψ	ψ
X	0	1	X	D	D’
D	ψ	Ψ	D	µ	ƛ
D’	ψ	Ψ	D’	ƛ	µ

ATPG path sensitization [9, 10] method generally comprises three phases Fault sensitization in which a stuck-at fault is activated by forcing a signal driving it to an opposite value from the fault value. Fault propagation in which the path is selected from the fault site to some primary output, where the effect of the fault can be observed for its detection. Line justification in which internal signal assignments previously made to sensitize a fault or to propagate its effect are justified by setting primary inputs of the circuit. In the second and third steps, we may find a conflict where a necessary signal assignment contradicts some previously made assignments this forces the ATPG algorithm to backtrack or backup. D_Algorithm Complete flow chart [2] given below in Fig (2).

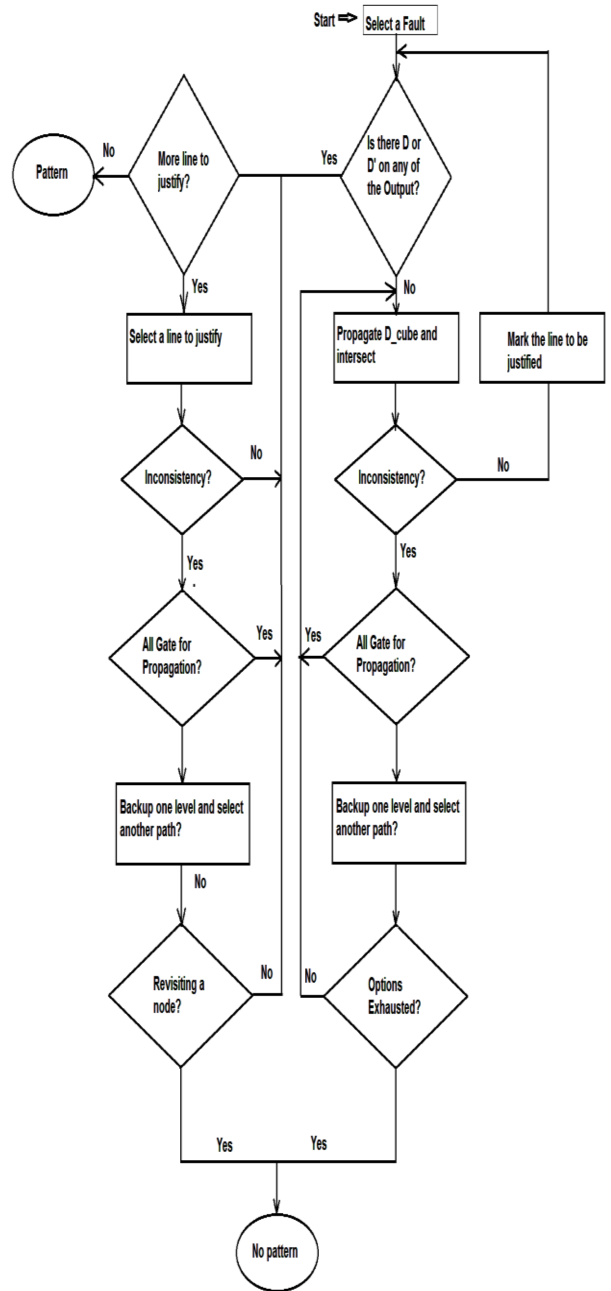

Fig 2: D_Algorithm flowchart.

2. THEORETICAL CALCULATION

Choosing stuck-at-0 fault at NOR gate output in given below circuit we will find input pattern for path sensitization. In which stuck-at-0 fault is activated by forcing a signal to a value 1 and these fault effect is propagated through path shown in fig [3] from output of NOR gate to output Z of circuit. Fault propagate to the output Z through gate no 2 ,3 and 4 i.e output of NOR gate given to the input of NAND gate and output of NAND gate connected to the First finding primitive D_Cube of fault for NOR gate and propagating Fault to the output of gate by calculating Propagation D_Cube of fault for gate 3 and 4 respectively. All calculations given bellow in D_Algorithm summary table with A, B, C, D, E are inputs and Z is final output of circuit.

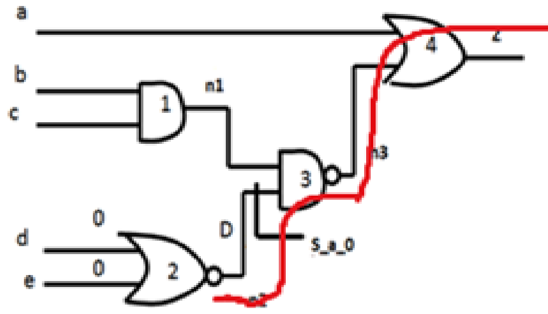

Fig 3: Design problem

D_Algorithm summary:

Table 4. D_Algorithm summary

	A	B	C	D	E	N1	N2	N3	Z
initial	X	X	X	X	X	X	X	X	X
Pdcf2	X	X	X	0	0	X	D	X	X
Pdc3	X	X	X	0	0	1	D	D1	X
Pdc4	0	X	X	0	0	1	D	D	D
Consi 1	X	1	1	X	X	1	X	X	X
	0	1	1	0	0	1	D	D	D1

Generated test pattern **01100**

3. SIMMULATION RESULT

In this part the emulation of D_Algorithm implemented. The implementation process will be carried out by writing PERL script to generate VHDL program files which are simulated on Xilinx 9.1.Text file containing net list of digital circuit with Fault and without fault processes and generated VHDL file and test bench file are simulated on Xilinx 9.1

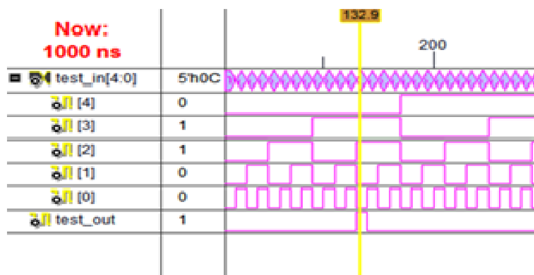

Fig 4: D_Algorithm simulation result

4. CONCLUSION

These paper focused on the designing ATPG algorithm like D_Algorithm for generating a minimum number of input pattern to test a circuit i.e. differentiate a fault free circuit from faulty circuit. Design D_Algorithm and generated a circuit and implemented it on FPGA Spartan II kit. minimum test pattern for particular combinational

5. REFERENCES

[1] Luca Mostardini, Luca Bacciarelli, Luca Fanucci, Lorenzo Bertini, Marco Tonarelli, Marco De Marinis, “FPGA –based Low-cost Automatic Test Equipment for digital integrated Circuits,” IEEE international workshop on intelligent Data Acquisition and Advanced computing system. Sept 2009

[2] F. Kocan, and D. Saab, “Concurrent d-algorithm on reconfigurable hardware, “Proceedings of the 1999 IEEE/ACM international conference on Computer-aided design, pp.152-156, 1999.

[3] S. Jamuna and V.K. Agrawal, “VHDL Implementation of BIST Controller,” in Proc. Of int. conf. on Advances in Recent Technologies in communication and computing. Pp. 188-190, 2011.

[4] E. J. McCluskey and F.W. Clegg, “Fault equivalence in combinational logic networks,” IEEE Tran computer vol. C-20, pp.1286-1293.

[5] I. Pomeranz and W. K. Fuchs, “A Diagonistic test generation procedure for combinational circuits based on test elimination,” in Proc. IEEE Asian Test Symp, pp.486-491.

[6] M. Abramovici, M. A. Breuer, and A. D. Friedman, Digital System testing and testable Design, IEEE press, 1994.

[7] T. Kirkland and M. R. Mercer, “A topological search algorithm for ATPG,” in proc. DAC-87, pp.502-508.

[8] Joe Dunbar “FPGA Based Design for Accelerated Fault-Testing of Integrated Circuits.” Bucknell University Jan 2010

[9] Miron Abeamovici, Melvin A.Breuer, Arthur D.Friedman, “Testing For Single Stuck At Faults” in Digital Systems and Testable Design. IEEE press New York.

[10] Michael L. Bushnell, Vishwani D. Agrawal , “Combinational Circuit Test generation” in Essential Of Electronic Testing For Digital Memory and mixed signal VLSI Circuits. Kluwer Academic Publisher New York, Boston, Dordrecht, London, Moscow.

[11] P. Ellervee, J. Raik, and V. Tihhomirov, “Fault emulation on FPGA: a feasibility study,” In Proceedings of the 21st NORCHIP Conference, Riga, Latvia, pp.92-95, 2003.

[12] V. D. Agrawal, D. H. Baik, Y. C. Kim, and K. K. Saluja, “Exclusive Test and its Applications to Fault Diagnosis," in Proc. 16th International Conf. VLSI Design, Jan. 2003, pp. 143-148.

[13] Yu Zhang “Diagnostic Test Pattern Generation and Fault simulation for Stuck-at and transition Faults,” a dissertation. August 4, 2012.

[14] P. Camurati, A. Lioy, P. Prinetto, and M. S. Reorda, “A Diagnostic Test Pattern Generation Algorithm," in Proc. International Test Conf., 1990, pp. 52-58.