International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 04 Issue: 08 | Aug -2017 www.irjet.net p-ISSN: 2395-0072

Design and Implementation of Test Vector Generation Using Random Forest Technique

Naveen T V[1], Dr. M V Latte[2], Mr. Sathish Shet[3], Mr. Shashidhara H R[4]

[1]M.Tech Student, VLSI Design & Embedded Systems, Dept. of ECE, JSSATE, Bengaluru, Karnataka, India
[2]The Principal, JSSATE, Bengaluru, Karnataka, India
[3]Asst. Professor, Dept. of ECE, JSSATE, Bengaluru, Karnataka, India
[4] Asst. Professor, Dept. of ECE, JSSATE, Bengaluru, Karnataka, India

---

Abstract: *Testing is an important part of any digital design process. For any quality products, it is important to determine the faults and to diagnose the fault in testing, so correctness and effectiveness will be the most important role in testing. Manual testing can be done for smaller designs, in which inputs can be given to the system by forcing the values and observing the output. As the complexity of the design increases the testing becomes tedious and hence automation is required. ATPG (Automatic Test Pattern Generation) is an automation method, which is used to find an input (or test) sequence applied to a digital circuit and enables the test equipment to compare between the correct and faulty circuit behavior caused by defects. These generated patterns are used to test integrated circuits after manufacture, and to assist for determining the reasons for failure. Usually the effectiveness of ATPG is calculated by the number of defects, or fault models which are determined and the total number of generated patterns used.*

Keywords: ATPG (Automatic Test Pattern Generation), DUT (Device Under Test), ATPG Algorithm.

1. INTRODUCTION

ATPG stands for automatic test pattern generation. As the name suggests, ATPG procedure involves generation of input patterns that can ascertain presence or absence of fault(s) at some location(s) in a circuit. There are several ways of ATPG.

- Fault simulation
- Sensitization–propagation–justification.

These techniques are based on Boolean logic manipulations and are most widely used. However, it must not be felt that only logic based schemes can be used for ATPG [3]. Here another interesting ATPG technique is that requires thermal imaging technique. Thermal imaging technique involves taking images of the silicon (of the circuit) and then drawing conclusions based on temperature profile of various regions of the silicon. In case of ATPG based on imaging technique, one needs to find an input pattern that results in difference of temperature at a location (which is being tested) with and without fault.

For example, when a net says A, has stuck-at-0 fault and primary inputs been applied such that net A gets 1, then temperature in A will be abnormally higher, due to high short circuit current. The advantage is, one need not have to worry about propagating the effect of the fault to primary outputs, as all internal points are observable in terms of temperature profile. So, sensitization is enough to generate a test pattern [6]. However, the instrumentation cost for thermal imaging is high. So, widely used ATPG algorithms are based on logic manipulations.

1.1 Types of ATPG Algorithms

Exhaustive: Testing by "exhaustive" technique implies applying all input combinations and checking for "functional" correctness. Exhaustive testing techniques are not used in ATPG because of complexity.

Random: Random ATPG basically involves three steps: Generate random pattern. Determine how many faults are detected by the random pattern (fault simulation) [7].Continue the above two steps till no (or few) new faults are detected by the random pattern.

2. DESIGN

The designing of the algorithm involves various techniques for the generation of test vector randomly for testing the digital circuits[1]. To test all the faults defined by the universal fault model in a combinational circuit with n PIs, we need to apply all 2^n possible input vector. The exponential growth of the required number of vectors limits the practical applicability of the testing method only to circuits with less than 20 PIs. Pseudo-random techniques for automatically generating simulation vectors are widely in use. Simulation tools may use random vector generators, and randomly generate **1**s and **0**s, collect a vector of inputs and apply them to the design and perform the simulation. Often, a situation arises where certain areas of code (or parts of the design) are not excited by the generated set of vectors. The basic block diagram is shown in figure 2.1.

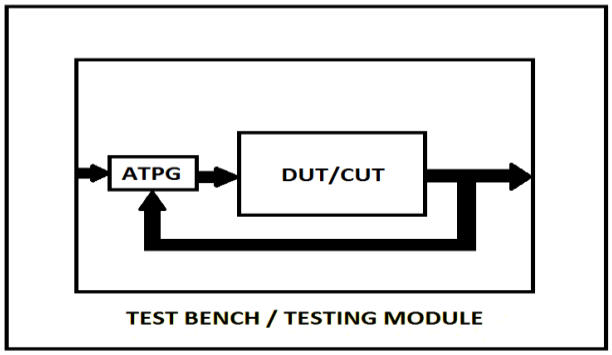

Figure 2.1: ATPG Module

At the inner core, it consists of the ATPG module and the DUT/CUT block. For simple designs ATPG block can be ignored but for complex designs they are bread and butter for testing. The ATPG block takes the input from the test bench/ testing module and produces the random test vectors. These test vectors are then used for the functionality testing of DUT[8].

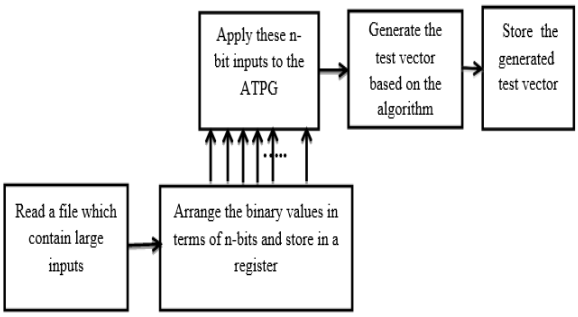

Fig 3.2: Block diagram for overall flow of test pattern generation

3. RESULTS AND DISCUSSIONS

The Test pattern generation unit has been designed and random test vector shave been obtained. The functional verification and validation are the two types of results that will be obtained. Let us see them in detail.

3.1 Functionality Verification:

Vector Generation: The figures show the results obtained for the designed structural method. As seen in the figure 3.1, the input considered is 8 for 'n' which generates 2^n vectors 256 sequences.

There can be a maximum of 2^n test vectors. Since only the test vectors which are greater than the input is stored we can observe numbers of test vectors are less than maximum possible vectors

The figure 3.2 shows the vectors generated in one-bit change. This helps the algorithm to generate the vectors in less number of clock cycles. As seen the vectors generated are random in nature.

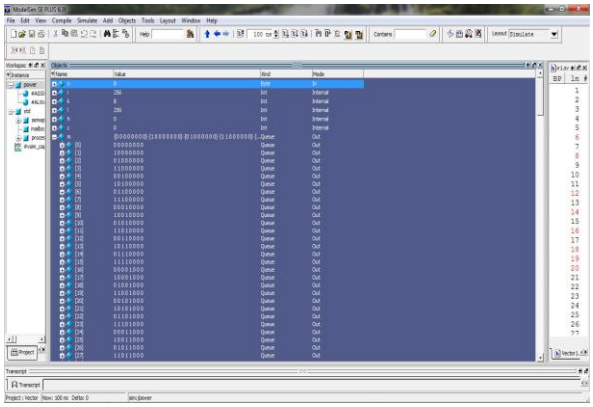

Figure 3.1: Simulation result of structural method vector generation

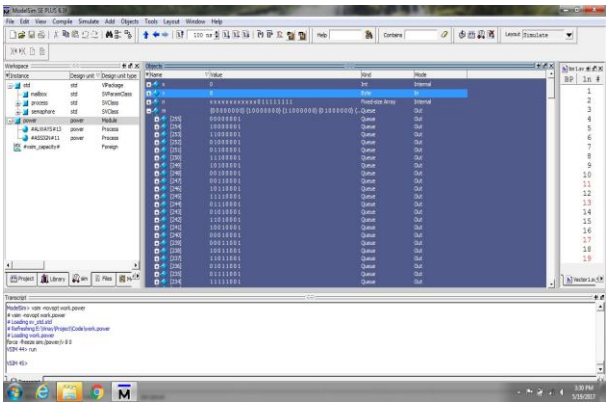

Figure 3.2: Simulation result of structural method one bit change vector

3.2. Validation of Combinational Circuits:

- **Multiplexer**

The multiplexer or MUX is a digital switch, also called as data selector. It is a combinational circuit with more than one input line, one output line and more than one select line. It allows the binary information from several input lines or sources and depending on the set of select lines, input line, is routed onto a single output line.

The figure 3.3 shows the block diagram of a multiplexer consisting of n input lines, m selection lines and one output line. If there are m selection lines, then the number of possible input lines is 2m. Alternatively we can say that if the number of input lines is equal to 2m, then m selection lines are required to select one of n (consider 2m = n) input lines.

This type of multiplexer is referred to as 2n × 1 multiplexer or 2n -to-1 multiplexer. For example, if one of the 4 input lines must be selected, then two select lines are required. Similarly, to select one of 8 input lines, three select lines are required.

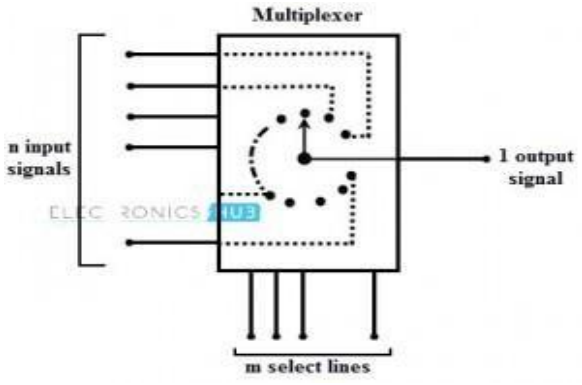

Figure 3.3: General working of MUX

- **8:1 Multiplexer**

An 8-to-1 multiplexer consists of eight data inputs D0 through D7, three input select lines S2 through S0 and a single output line Y. Depending on the select lines combinations, multiplexer decodes the inputs.

The figure 3.4 shows the block diagram of an 8-to-1 multiplexer with enable input that enable or disable the multiplexer. Since the number data bits given to the MUX are eight then 3 bits (23=8) are needed to select one of the eight data bits.

The figure 3.5 shows the validated output of the DUT 8:1 MUX linked to the vector generation code. Here the select line is given as 010 whose decimal value is 2, hence 3rd line in output MUX is validated. Any number of DUT's can be linked to the ATPG program for validation.

The figure 3.6 shows the output of the 8:1 MUX when the program has error in it, so the program can be altered to generate proper output.

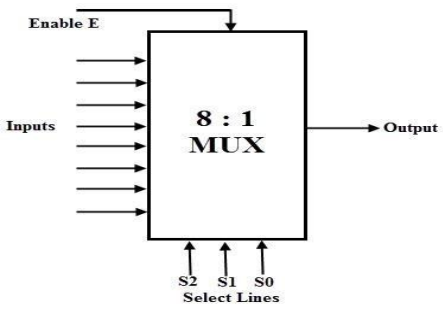

Figure 3.4: Block diagram of 8:1 MUX

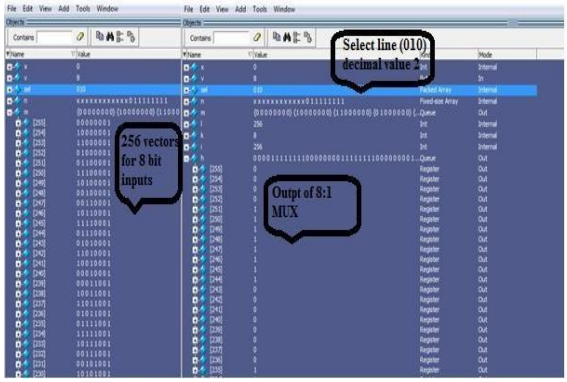

Figure 3.5: Valid 8:1 MUX output

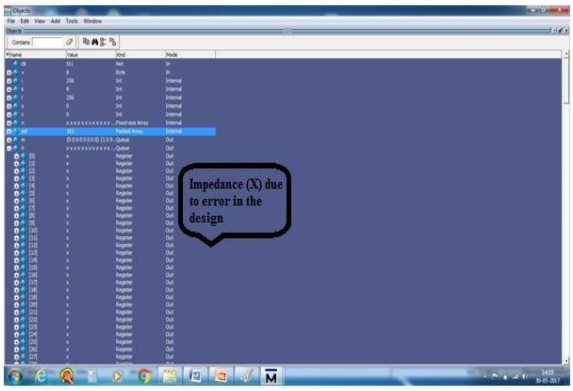

Figure 3.6: Invalid 8:1 MUX output

3.3 Validation of Sequential Circuits

- **D Flip Flop**

D flip – flops are also called as "Delay flip – flop". They are used to store 1 – bit binary data. They are one of the widely used flip – flops in digital electronics. Apart from being the basic memory element in digital systems, D flip – flops are also considered as Delay line elements and Zero – Order Hold elements.

D flip – flop has two inputs, a clock (CLK) input and a data (D) input and two outputs; one is main output represented by Q and the other is complement of Q represented by Q'. The symbol of a D flip –flop is shown in fig 3.7.

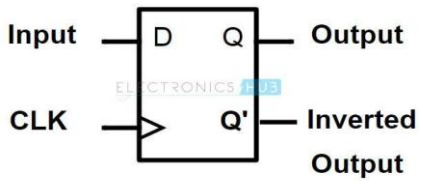

Figure 3.7: block diagram of D Flip-Flop

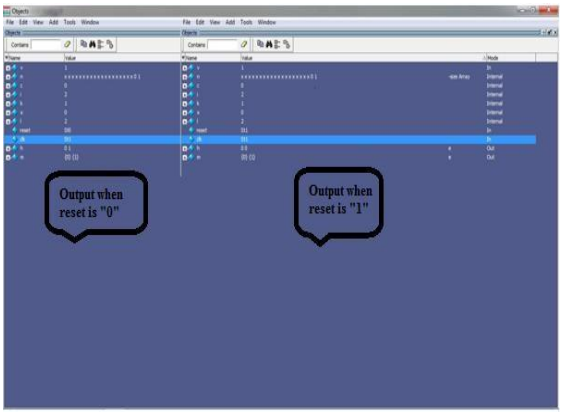

Fig 3.7: Output of D FLIP FLOP design

4. CONCLUSION

Testing is an important part of any digital design process. Testing is used in semiconductor technology to distinguish between the correct and faulty behavior. Testing and its application and has undergone tremendous development and growth. ATPG is one of the most popular techniques of testing VLSI design. This project is designed for two methods of random test vectors generation. Verified the functionality of the designed ATPG module. Validated the Verilog design of 8:1 multiplexer and D Flip Flop using ModelSim. Can improve the design to locate the actual fault. Randomization can be increased up to N states. Synthesize and Physical design can be implemented for the design.

5. REFERENCES

[1] Shashidhara, H. R. "Test vector Generation using different logic regression method". Applied and Theoretical Computing and Communication Technology (iCATccT), 2016 2nd International Conference on. IEEE, 2016.

[2] S Hemalatha, "Automatic Test Generation for Digital Circuits", International Journal of Scientific & Technology Research Volume 3, Issue 2, February 2014.

[3] John Sunwoo, Vishwani D. Agrawal," A new ATPG algorithm for 21st century", Dept. of Electrical and Computer Engineering Auburn University

[4] Raghuram S. Tupuri and Jacob A. Abraham, "A Novel Functional Test Generation Method for Processors using Commercial ATPG", Advanced Micro devices, University of Texas at Austin

[5] Tracy Larrabee, "A Framework for Evaluating Test Pattern Generation Strategies", International Test Conference, March 1990

[6] Tracy Larrabee, "Efficient Generation of Test Pattern Using Boolean Difference", International Test Conference, March 1990.

[7] Yu Zhang, "Diagnostic Test Pattern Generation and Fault Simulation for Stuck-at and Transition Faults", A dissertation submitted to the Graduate Faculty of Auburn University.

[8] Abramovici, Miron, Melvin A. Breuer, and Arthur D. Friedman. Digital systems testing and testable design. Vol. 2. New York: Computer science press, 1990.

[9] ASIC Lab Manual Revision 4.0 IES132 RC132 EDI132 Developed by University Support Team Cadence Design Systems, Bangalore.

[10]http://en.wikipedia.org/wiki/Automatic_test_pattern_generation

[11] ModelSim® Tutorial, Software