Novel Implementation of Low Power Test Patterns for In Situ Test

K.Ramya¹, Y.N.S.Vamsi Mohan², S.V.S.M.Madhulika³

¹M.Tech Student, Department of ECE, Bonam Venkata Chalamayya Institute of Technology & Science, Amalapuram, India.
²Associate Professor Department of ECE, Bonam Venkata Chalamayya Institute of Technology & Science, Amalapuram, India.
³Assistant Professor Department of ECE, Bonam Venkata Chalamayya Institute of Technology & Science, Amalapuram, India.

Abstract: Test vector generation, its application to CUT and its response analysis are the tasks done by the In Situ Test. A new and efficient approach for the Generation of all one bit changing random input patterns for in situ test is developed in this paper using counter with gray. In this proposing technique, the counter with gray is used to generate all the $2^n$ one bit change test vectors to overcome the limitations of existing two counters viz., Johnson counter and scalable SIC counter of n-bit size, because they consists of $(2^n - 2n)$ unused test patterns. So this approach of in situ testing can be done with less power dissipation because switching power dissipation is reduced as we are applying one bit change binary test patterns. The developed test vector generation is suitable for both the test per clock and test per scan based BIST systems. We have carried the simulation and verified the results using ISE simulator and synthesis is done on the XILINX ISE.

Index Terms—IN SITU test, low power, one bit change binary patterns, test vector generator (TVG), Johnson counter, scalable SIC counter, counter with gray.

I. INTRODUCTION

The major issues that are needed to be considered in VLSI are its performance, size, speed and power consumption. Usually circuits will require more power to operate during testing when compared to operating normally. So huge amount of power usage at the time of testing results in less yield of the chips. The grown heat because of additional power consumption can make our VLSI chips less reliable due to electro-migration at the time of testing. In worst cases excessive power usage at testing may also cause our chips to burn. The sources of power dissipation are switching, short circuit, leakage and static power dissipations. The key principles for low power design are - to use the lowest possible supply voltage, to use the less geometrical area, Power management by not connecting the power source when the system is not used, to reduce the switching activity. In them switching power dissipation is more so to decrease it we need to reduce switching activity. For that sake we are creating the test vectors with only one bit change for one test pattern to the other. So here we are developing a new approach of reducing the power usage during testing of a circuit using in situ test as this in situ test produces all the test vectors with only one input change from one another.

BIST is a special circuit designed generally for ICs within it - whereby the actual IC tests itself by using this special circuitry. If the entire circuitry performing the test is contained within an IC, we call it self-test, in situ test, or built-in self-test.

The fundamental idea of In Situ Test, in its most simple form, is to design a circuit so that the circuit can test itself and determine whether it is ‘good’ or ‘bad’.

The in situ applies the binary patterns for the circuit under test and the obtained response is checked with the good expected response. If the CUT gives the correct response for all the test vectors then the CUT is said to be fault free otherwise we say it is a faulty circuit.

Need for In Situ Test:

Presently we are having the systems with heavily integrated several-layer boards with IC’s in it so accessing the individual IC’s from the system are really impossible and also cost of testing increase a lo
According to rule of ten. As rule of ten says if a chip fault is not caught by chip testing, then to find the fault at the PCB level costs 10 times more compared to as at the IC level. In the similar way, if a board fault is not caught by PCB testing, and then finding it at system level costs 10 times higher as compared to finding at the board level. So some people say that the rule of ten should be called as the rule of twenty, because the chips, boards and systems are becoming even more complex than at the time when the rule was first stated.

External tester problems

The large number of gates in VLSI circuits has pushed computer automatic-test-generation times to weeks or months of computation. The numbers of test patterns are becoming too large to be handled by an external tester and this resulted in high computation costs and has outstripped reasonable available time for production testing. Due to heavy number of gates in VLSI circuits the computer automatic test vector generation times are taking nearly weeks or months for computation. The external tester was unable to handle the large number of test and due to this we are requiring high computation costs and also exceeds available time for production testing.

The ratio of gate count to pins in IC’s is becoming a problem

As the number of gates per an IC are increasing, the pins are not increasing in that number so we are unable to access all the gate input and output nodes. This results in testing of internal nodes very difficult as the gate inputs are not controllable through any input pin and are not observable by any output pin.

Hardware Test Vector Generators for in situ test

The following hardware test vector generators are used for in situ test. They are

1. ROM: in this method we store a good test-vector set (from an ATPG program) in a ROM which was placed on the IC itself, but this requires usually more chip area so it was not a preferable approach for input pattern generation.

2. LFSR: a linear feedback shift register (LFSR) is used to generate pseudo-random tests. This usually requires a sequence of 1 million or more tests to obtain good fault coverage and it uses very little hardware.

3. Binary Counters. A binary counter can generate an exhaustive test sequence, but this can use too much of test time if the number of inputs are large. The binary counter requires more hardware than the typical LFSR pattern generator.

4. Modified Counters. Modified counters have also been successful as test-pattern generators, but they also require long test sequences.

5. LFSR and ROM. One of the most effective approaches is to use an LFSR as the primary test vector generator, and then generate test-vectors with an ATPG program for the faults that are missed by the LFSR sequence. These few additional test-patterns can either be stored in a small ROM on the chip for a second test epoch

6. Cellular Automaton. In this approach, each pattern generator cell has a few logic gates, a flip-flop, and connections only to neighboring gates. The cell is replicated to produce the cellular automaton.

ILEXISTING COUNTERS FOR MSIC-TPG SCHEME

Reconfigurable Johnson Counter

Based on the scan length the existing technique uses two kinds of one bit change test vector generators to generate several one bit change vectors. They are Reconfigurable Johnson counter and Scalable SIC counter. For small scan length, we are using an L-bit reconfigurable Johnson counter to produce 2L one bit change test vector sequence in time domain. This can be used worked in any of the three modes.

1) Initialization: This counter will produce zeroes on all d-flip flops after L CLK cycles when we applied logic ‘1’ on RJ_MODE and logic ‘0’ on Init.
2) Circular shift register mode: In this mode, each stage of it will produce Johnson codeword after L CLK cycles When RJ_Mode and Init are kept at logic ‘1’.
3) Normal mode: After initialization this counter will produce 2L different one bit change vectors When RJ_Mode is set to logic 0 for 2L CLK cycles.

Here we will use this L-bit reconfigurable Johnson counter in normal mode which produces 2L one bit change vectors. Usually when we take an L-bit counter we will have 2^L combinations so by using this counter we are having (2^L-2L) unused test vectors.
Scalable SIC Counter

When the scan chain length is larger, we are using another counter “scalable SIC counter” in order to produce n bit change test vectors. As shown in Fig. 3(a), it has a k-bit adder which changes its output for every clk cycle and it is clocked by the rising SE signal, a k-bit subtractor is clocked by CLK2, an M-bit shift register clocked by test clock CLK2, and we are using k multiplexers.

The simulated result of the scalable SIC counter is shown in Fig. 3(b). this counter also can be worked in three modes.

A. When we set SE to logic ‘0’ then the count value in the adder is moved to the subtractor. When SE = 1, the value in the k-bit subtractor will be decremented from the initial stored value gradually to zero.

B. If SE = 1 then the MSB bit of the k-bit subtractor is moved to the M-Johnson that is M-bit shift register LSB.

C. Thus, the required 1s (0s) will be shifted into the M-bit shift register after L CLK2 cycles, and the generated 2L unique one bit change vectors are applied into different scan chains.

III. PROPOSED EFFICIENT COUNTER FOR MSIC-TPG SCHEME

Counter with gray

The efficient way to generate all the 2L one bit change test vectors is to use L-bit counter along with binary to gray converter. This efficient special counter consists of an L-bit normal counter followed by a binary to gray code converter. The L-bit normal counter generates all the 2L combinations in binary which were not one bit change vectors. So to make them one bit change vectors we are giving the outputs of normal counter to a binary to gray code converter as the gray code numbers will have only one bit change between two successive numbers. So the output from this counter are 2L one bit change test vectors. Its block diagram and corresponding waveforms are shown in fig 4(a) and 4(b).
Now here in this project we are using the above mentioned counters in order to develop one bit change Test Vector Generator for both the Test Per Clock and the Test Per Scan based In Situ Test System. So using this test vector generator the switching power dissipation is reduced during testing so this test of using the one bit change test vectors is termed as ‘Low Power In Situ Test’.

IV. MSIC-TPGs FOR BIST SYSTEMS

**MSIC-TPGs for Test-per-Clock Schemes**

Usually, in a test-per-clock IN SITU TEST system, some new set of faults is tested during every clock period. Consider a BIST pattern sequence of 10 million vectors, applied at the system operating speed of 200 MHz. Testing takes only 10,000/200*10^6=0.05s. The lengthy computation time worsens as the BIST pattern sequence increases, because the entire sequence must be simulated for the good machine and all failing machines.

The one bit change test vectors for test per clock scheme are illustrated in fig.5. here we are using LFSR as seed generator which is of m-bit size and we are using any of the above mentioned counters which produce one bit change vectors of n-bit. So the outputs from the seed generator and the counter are given to the XOR gates arranged as n × m SRAM-like grid structure. So the output of the XOR gates can be used as inputs for CUT. So when we are applying as column wise we have m number of n-bit test vectors.

The test procedure is as follows.

1) The seed generator will produce a new seed value for every CLK1 cycle.
2) The counter produces a new one bit change for every CLK2 cycle.
3) Repeat 2 until all the different one bit change patterns are produced by a counter.
4) Repeat 1–3 until that test length to cover all the expected faults has been achieved.
In a test-per-scan BIST system, each new set of faults that is tested requires one clock to conduct the test and a series of shifts of the scan chain to complete that test and read out all of the test results. Test-per-scan, therefore, takes significantly more time than a test-per-clock method to detect the same number of faults in a given circuit. The advantage of test-per-scan systems over test-per-clock systems is that a judicious combination of scan chains and a MISR can lead to a significantly smaller MISR than in a test-per-clock system.

The MSIC-TPG for test-per-scan schemes is illustrated in Fig. 7. The clk and control circuit takes the clk, rst, tn_mode and produces CLK1, CLK2, Init, RJ_MODE. The inputs of the XOR gates come from the seed generator and the counter and their outputs are applied to scan chains, respectively. The outputs of the seed generator and XOR gates are given as inputs to the CUT and for MISR also to produce the response at primary outputs and good expected response respectively. The primary outputs of CUT are given to the MISR and then it tests this with the good expected response and generates an output c_f signal whether there is fault or not.

The test procedure is as follows.

1) The seed circuit generates a new seed for every clk1 cycle.
2) The counter will generate a one bit change test vector by clocking CLK2 one time.
3) Repeat 2 for all the unique one bit change vectors are generated.
4) The outputs of seed circuit and counter are given to xor gate to form scan chain
5) Repeat 1–4 to produce the test length such that an expected fault coverage has been achieved.
V. CONCLUSION

This paper has proposed a low-power test vector generation for In Situ Test as we are using one bit change test vectors to reduce the switching power dissipation. Combined with the proposed counter with gray and the existing Reconfigurable Johnson counter or scalable SIC counter, the MSIC-TPG can be easily implemented. We have developed it for both test-per-clock and test-per-scan schemes. For a test-per-clock scheme, the MSIC-TPG uses its one bit change vectors as inputs to the CUT with the SRAM-like grid. For a test-per-scan scheme, the MSIC-TPG converts a one bit change vector to low transition vectors for all scan chains and these are applied to CUT along with the outputs of seed circuit. The simulation results shows that how the patterns are generated and applied for CUT and CUT’S response is analyzed to make a decision whether this circuit is fault free or not. So this paper presents the novel implementation of all the one-bit change approach based test vector generator by using verilog language. Synthesizing and simulation of the code is carried out on Xilinx - Project Navigator, ISE suite.

REFERENCES

[1] Feng Liang, Luwen Zhang, Shaochong Lei, Guohe Zhang, Kaile Gao, and Bin Liang “Test Patterns of Multiple SIC Vectors: Theory and Application in BIST Schemes” in IEEE transactions on very large scale integration (vlsi) systems, Vol. 21, No. 4, April 2013


