International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 6 (2015) pp. 15293-15304
© Research India Publications
http://www.ripublication.com

Low Power Analysis and Design of Efficient Mlat Fault Diagnosis Algorithm With Test Pattern Generation

Dr.T.Jaya
Professor, C.S.I Institute of Technology, Thovalai,
KanyaKumari,jayacsiramesh@gmail.com
Dr.C. Jeya Chandiran
principal, Udaya School of Engineering, KanyaKumari
M.Monisha Sibiah, PG scholar, C.S.I Institute of Technology,
Thovalai, Kanya Kumari

Abstract

Devices are manufactured using VLSI technology. In VLSI technology billion of transistors are integrated in a single chip. During integration failure may occur. Fault diagnosis plays an important role to find faults in the chip. Single fault is identified using single fault based diagnosis method such as the single location at a time (SLAT). This method become invalid if multiple faults, fault masking and reinforcing may exists. The existing method use fault element and fault element graph to diagnose faults occurred at multiple locations. This method includes the effect of fault masking and fault reinforcing which are the problems in diagnosing multiple faults. This method construct FEG for all failing pattern and fault element scores the solution is identified. All candidate locations are ranked to constitute the final diagnosis result. This may lead to many solutions to be identified which cause long run time. The existing algorithm only based on failing pattern. The proposed system use fault diagnose algorithm which is used to reduce long run time by introducing limitation on number of solutions. The new fault diagnose based algorithm is based on both passing and failing pattern. The proposed system has the advantage of better accuracy and reduced delay. Johnson counter is used to generate the test pattern which is used to analyze the circuit.

Introduction

Testing is important parameter in VLSI technology. Many challenges are imposed on tools and methodologies used to design and test complex VLSI circuits. The most important issues in the development process of an integrated circuit during testing are manufacturing yield, product quality, and test cost. To describe manufacturing defects and their fault models in the circuits, test patterns are generator which is applied to the circuit under test (CUT) and to detect faulty circuits can be done either externally using automatic test equipment (ATE) or internally using built-in self-test (BIST) [1].

The CUT may contain a resistive-open defect is defined as an imperfect circuit connection that can be modeled as a defective resistor between the circuit nodes that should be connected. A defective wire that can be modeled by a resistive open defect (Rdef). Examples of resistive-open defects are thin wires[2].

Determining the location is much simpler, and can be done in a piecemeal fashion by analyzing failing patterns and building up a composite picture of the defect's whereabouts. This diagnostic technique is called single location at a time (SLAT) because it uses only those patterns during which the defect affected only a single location, be that a pin or a net[3].

Diagnosing multiple combinational logic faults can be classified into two categories: diagnosticpattern-based diagnosis methods [4]–[7] and manufacturingtest-pattern-based diagnosis methods [8]–[15]. In the first category [4]–[7], failure chips' failing responses under manufacturing test patterns are analyzed to obtain the initial candidate faults. When testing a chip, if the chip's response under a test pattern is different from the expected good machine response, the test pattern is a failing pattern, and the response is called the failing response.

For the manufacturing-test-pattern-based diagnosis methods, no diagnostic patterns are generated and used. Failure chips' failing responses under manufacturing test patterns are compared with potential faults' failing responses to find the candidate faults. In [8]–[15], fault simulations are applied to obtain the failing responses of potential faults. In [14] and [15], the X-fault model is used to represent the unknown behavior of potential faults. The faults that can produce more matched failing outputs and cause less passing outputs to fail are ranked higher in the reported candidate fault list.

Overview

The work is splitted into two phases. They are,

1) Generation of test pattern
2) Circuit analysis

Block Diagram For Generation of Test Pattern

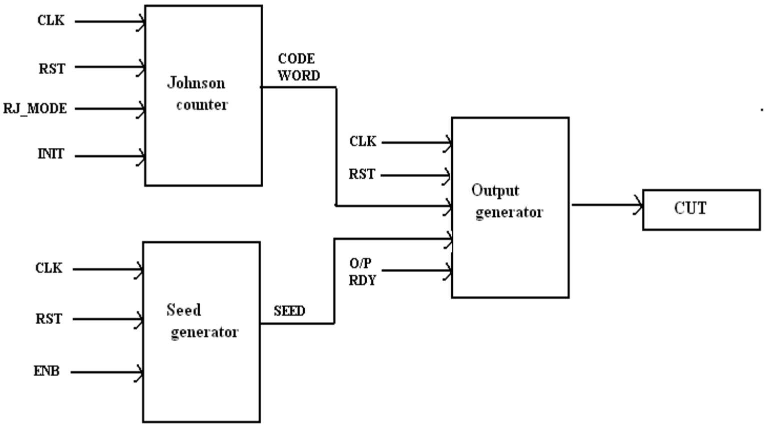


Figure 1: Test pattern generator block diagram

Steps in producing the outputs are:-

1) The seed generator generates a new seed by clocking CLK one time using enable seed.
2) RJ_mode is set to '0'. The Johnson counter will operate in circular normal mode and generate a Johnson vector by clocking more than one time.
3) After that RJ_mode and Init mode are set to 1, the Johnson counter will operate as a circular shift register and produce n codewords by clocking CLK n times.
4) If 2n Johnson vectors are generated and then XOR the output of seed generator with the output of Johnson counter.
5) The outputs are given to the circuit under test (CUT).
6) The procedure are repeated until the fault coverage is achieved.

A. Analysis of Johnson Counter

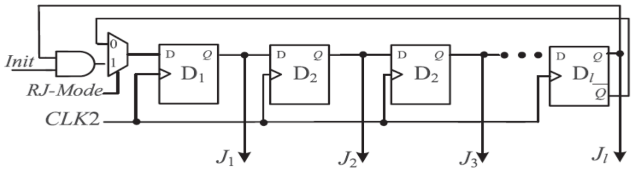


Figure 2: Johnson counter

Johnson counter will operate in three modes of operation. They are

1) Initialization
2) Circular shift register
3) Normal mode

1) Initialization :
   In this mode of operation RJ_mode is set to logic 1. Init is set to logic 0. Then the Johnson counter will produce 0 as the output for all stages by clocking CLK more than n times.
2) Circular shift register:
   In this mode of operation RJ-mode are set to logic1. Then the Johnson counter will produce Johnson codeword by clocking CLK n times.
3) Normal mode:
   RJ_mode is set to logic, the reconfigurable Johnson counter will produce 2n unique vector.

B. Seed Generator

Seed generator has three inputs. They are clock, reset enable. The input signal clock is always set to 1. If the input signal reset is set to 1, it will display all the output as 0. If the input signal reset goes to 0 but enable signal will rise to high value then it will generate the output as equal to the output of linear feedback shift register output. Then this output and Johnson counter output are given to the XOR gate to produce the test pattern.

Circuit Analysis

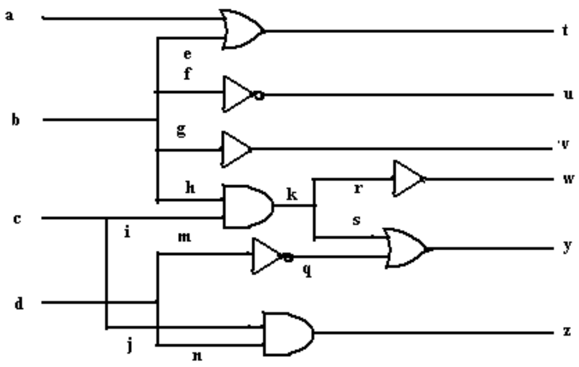
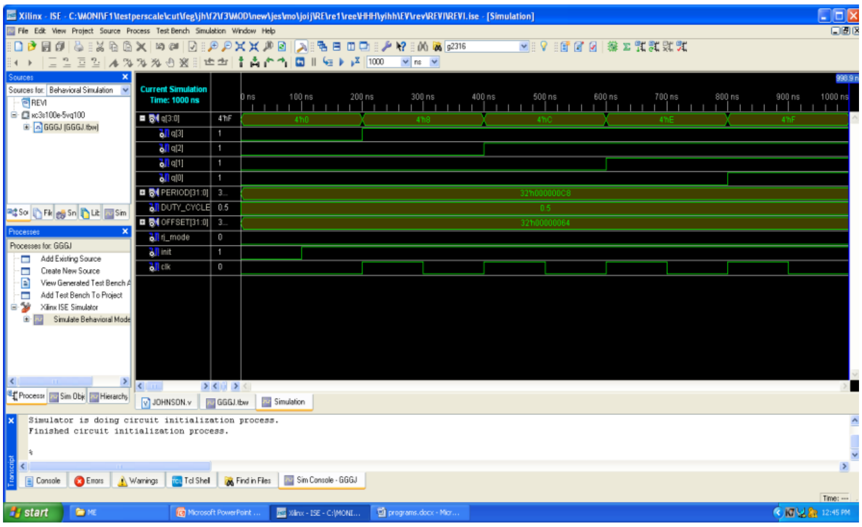


Figure 3: Circuit Under Test

The circuit has AND,NOT,OR gates. To identify faults in the circuit, it will be explained by the use of fault element method. Consider a circuit with three fault locations q, b, and c. The three faults produce three failing patterns. The good machine value of each location is written aside the location label. The FEG of each failing pattern is given below the circuit. In the FEGs, each vertex represents a fault element, and each directed line represents the relation between corresponding fault elements, which will be explained next. The score of a fault element is written below the fault-element label in the vertex.

The circuit was tested using 3 different test pattern and the faults are analyzed and the graph was plotted. Three test patterns are P1=abcd(0010) ,P2=abcd(0110) ,P3=abcd(1110)

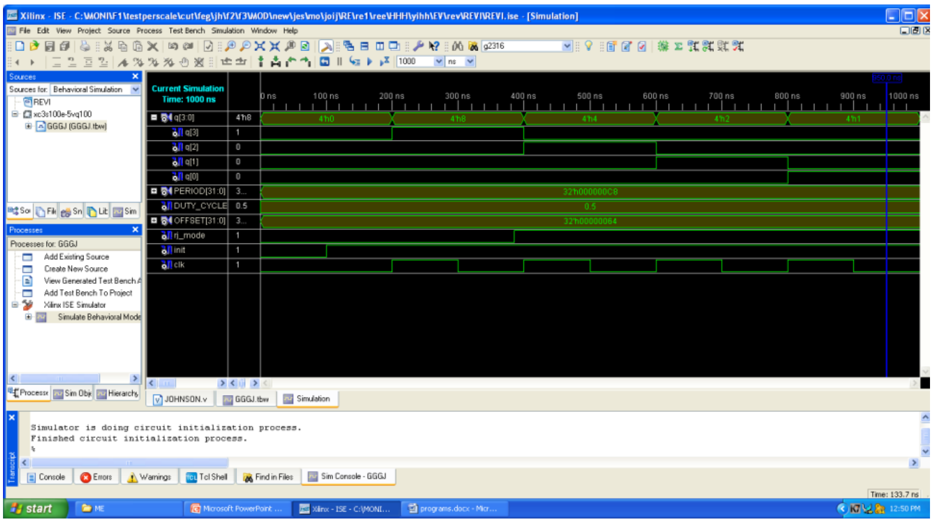
Hargware Implementation and Results



```
Simulator is doing circuit initialization process.
Finished circuit initialization process.
```

Figure 4: Johnson Counter Normal Mode Output

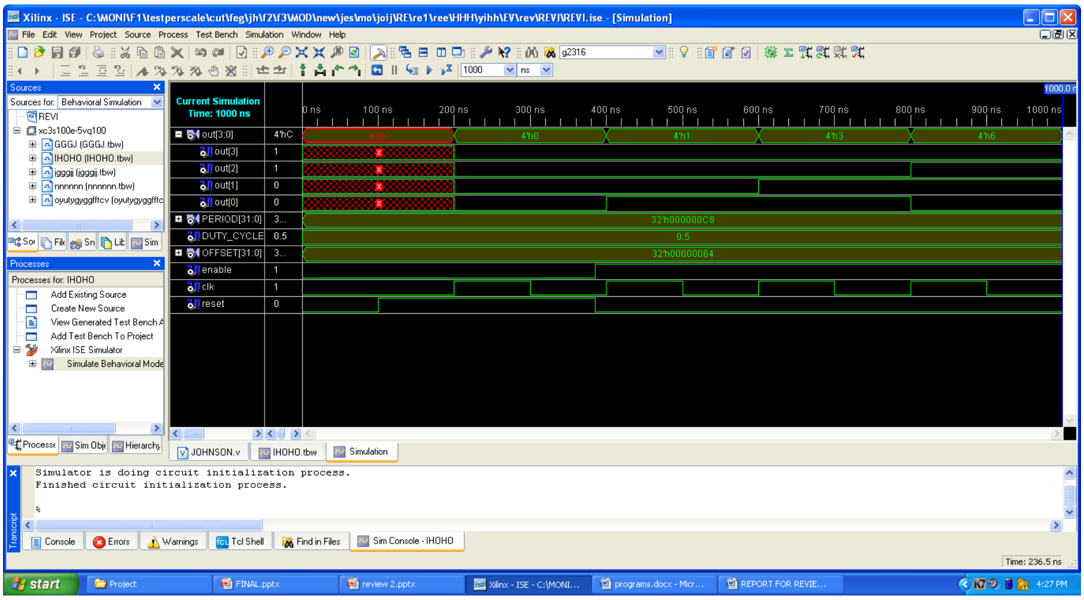
Johnson counter normal mode output is obtained by setting rj-mode value as 0 and init mode value as 1. The MSB value is set to 1 by XOR ing the 3rd bit and the 1st bit value. This value is set throughout the operation. And it passed to the prior MSB bit. The circuit will generate the output with delay 4.063ns



```
Simulator is doing circuit initialization process.
Finished circuit initialization process.
```

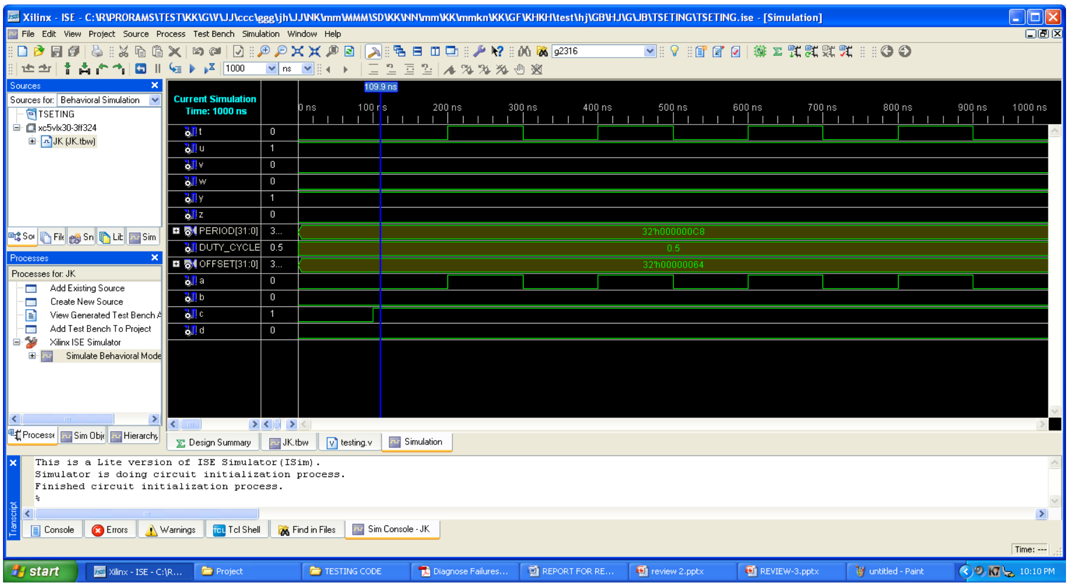
Figure 5: Johnson Counter Circular Mode Output

The Johnson counter will operate in circular mode by setting the rj-mode value and init value as 1. In this mode the MSB is set to high value at first clock cycle. At the rising edge of second clock cycle the MSB value was changed to 1 and prior MSB bit is changed to the value 1. This process is continued till all the bit value is changed to 1. It will generate the output with delay 4.063ns



```
Simulator is doing circuit initialization process.
Finished circuit initialization process.
```

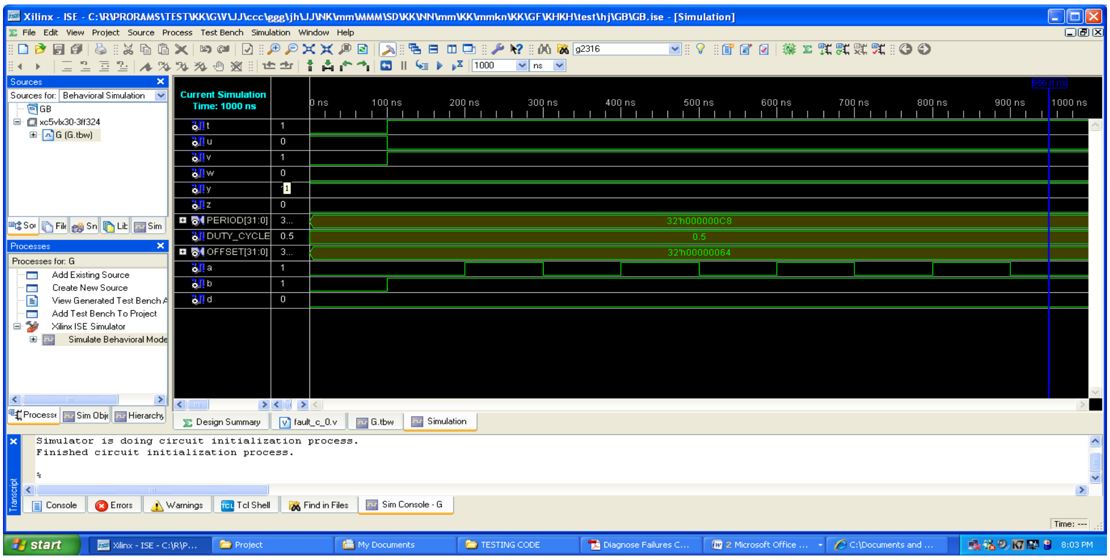
Figure 6: Seed Generator Output



```
This is a Lite version of ISE Simulator(ISim).
Simulator is doing circuit initialization process.
Finished circuit initialization process.
```

Figure 7: Output For Pattern 0010

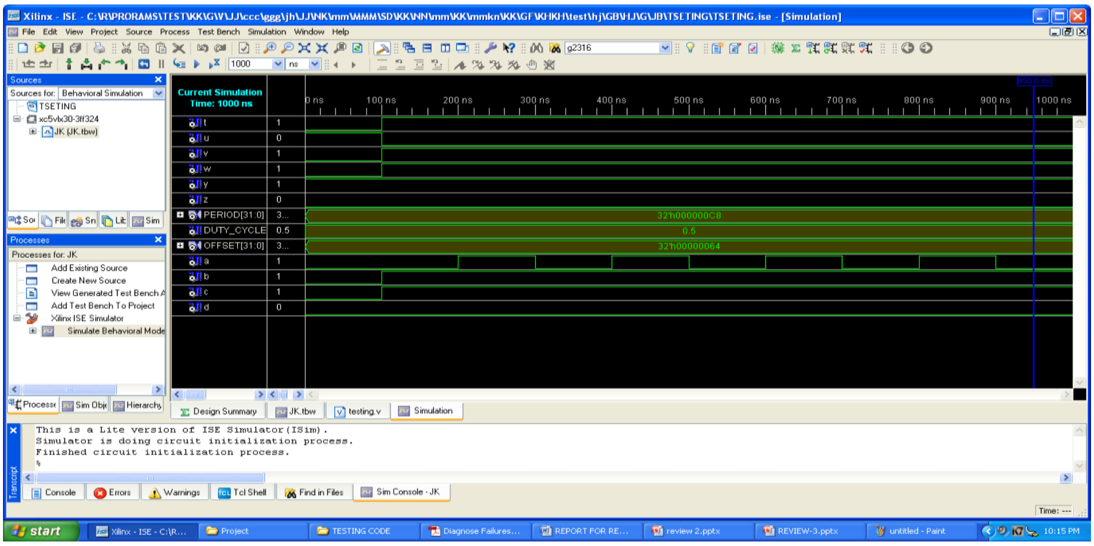
The above figure shows the output for pattern 1. The pattern 1 value is 0010. This is applied to the input line abcd. The t output is produced by input a and b. These two input values are given to OR gate and it wll produce the output at t. The u value is obtained by passing the b value through a NOT gate. The v value is same as the b value. The w output is obtained by passing input b and c through a AND gate. The y output is obtained by passing d value via NOT gate and combinig this with a w value by the use of OR gate. The z output is produced by passing c and via AND gate. Then the circuit will produce the output as 010010at port tuvwyz. The circuit will produce the delay as 5.934ns



```
Simulator is doing circuit initialization process.
Finished circuit initialization process.
```

Figure 8: Faulty Output For Pattern 0010.

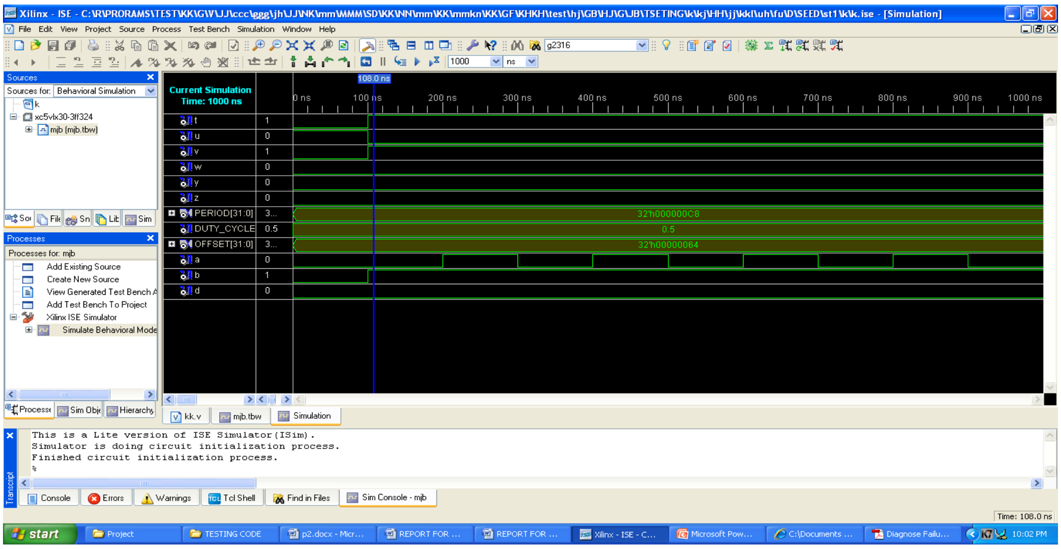
The above figure shows the faulty output for pattern 1 (0010). Faults are injected in b,q and c. This is applied to the input line abcd. The t output is produced by input a and b. These two input values are given to OR gate and it wll produce the output at t. The u value is obtained by passing the b value through a NOT gate. The v value is same as the b value. The w output is obtained by passing input b and c through a AND gate. The y output is obtained by passing d value via NOT gate and combinig this with a w value by the use of OR gate. The z output is produced by passing c and via AND gate. Due to the fault in the circuit the output is changed to 101110 for the same input 0010. The circuit will produce the delay as 5.934ns


Xilinx - ISE - C:\R\PRORAMS\TEST\KK\G\V\JJ\ccc\ggg\jh\JJ\NK\mm\MMM\SD\KK\NN\mm\KK\mmkn\KK\GF\KHKH\test\hj\GB\HJ\G\JB\TSETING\TSETING.ise - [Simulation]
File Edit View Project Source Process Test Bench Simulation Window Help
Sources for: Behavioral Simulation
TSETING
xc5vlx30-3ff324
JK (JK.tbw)
Processes for: JK
Add Existing Source
Create New Source
View Generated Test Bench As HDL
Add Test Bench To Project
Xilinx ISE Simulator
Simulate Behavioral Model
Current Simulation Time: 1000 ns
t 1
u 0
v 1
w 1
y 1
z 0
PERIOD[31:0] 3...
DUTY_CYCLE 0.5
OFFSET[31:0] 3...
a 1
b 1
c 1
d 0
32'h000000C8
0.5
32'h00000064
0 ns 100 ns 200 ns 300 ns 400 ns 500 ns 600 ns 700 ns 800 ns 900 ns 1000 ns
Design Summary
JK.tbw
testing.v
Simulation
Console
Errors
Warnings
Tcl Shell
Find in Files
Sim Console - JK
start
10:15 PM

```
This is a Lite version of ISE Simulator(ISim).
Simulator is doing circuit initialization process.
Finished circuit initialization process.
```

Figure 9: Output For Pattern 0110

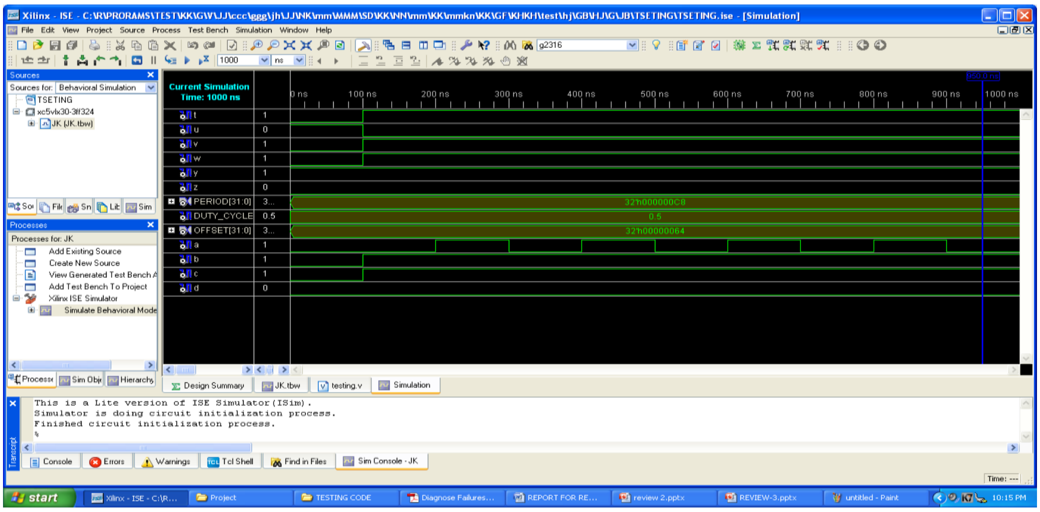
The above fig9 shows the output for pattern 2 (0110). The input value 0110 is applied to the circuit This is applied to the input line abcd. The t output is produced by input a and b. These two input values are given to OR gate and it wll produce the output at t. The u value is obtained by passing the b value through a NOT gate. The v value is same as the b value. The w output is obtained by passing input b and c through a AND gate. The y output is obtained by passing d value via NOT gate and combinig this with a w value by the use of OR gate. The z output is produced by passing c and via AND gate. Then the circuit will produce the output as 101110. The circuit will produce the delay as 5.934ns


Xilinx - ISE - C:\R\PRORAMS\TEST\KK\G\V\JJ\ccc\ggg\jh\JJ\NK\mm\MMM\SD\KK\NN\mm\KK\mmkn\KK\GF\KHKH\test\hj\GB\HJ\G\JB\TSETING\k\kj\HH\jj\kkl\uh\u\D\SEED\st1\k\k.ise - [Simulation]
File Edit View Project Source Process Test Bench Simulation Window Help
Sources for: Behavioral Simulation
k
xc5vlx30-3ff324
mjb (mjb.tbw)
Processes for: mjb
Add Existing Source
Create New Source
View Generated Test Bench As HDL
Add Test Bench To Project
Xilinx ISE Simulator
Simulate Behavioral Model
Current Simulation Time: 1000 ns
108.0 ns
t 1
u 0
v 1
w 0
y 0
z 0
PERIOD[31:0] 3...
DUTY_CYCLE 0.5
OFFSET[31:0] 3...
a 0
b 1
d 0
32'h000000C8
0.5
32'h00000064
0 ns 100 ns 200 ns 300 ns 400 ns 500 ns 600 ns 700 ns 800 ns 900 ns 1000 ns
kk.v
mjb.tbw
Simulation
Console
Errors
Warnings
Tcl Shell
Find in Files
Sim Console - mjb
Time: 108.0 ns
start
10:02 PM

```
This is a Lite version of ISE Simulator(ISim).
Simulator is doing circuit initialization process.
Finished circuit initialization process.
```

Figure 10: Faulty output for pattern 0110

The above fig10 shows the output for pattern 2 (0110). The input value 0110 is applied to the circuit. This is applied to the input line abcd. Faults are injected in b,q and c. The t output is produced by input a and b. These two input values are given to OR gate and it wll produce the output at t. The u value is obtained by passing the b value through a NOT gate. The v value is same as the b value. The w output is obtained by passing input b and c through a AND gate. The y output is obtained by passing d value via NOT gate and combinig this with a w value by the use of OR gate. The z output is produced by passing c and via AND gate. Then the circuit will produce the output as 101110. The circuit will produce the delay as 5.934ns



```
This is a Lite version of ISE Simulator(ISim).
Simulator is doing circuit initialization process.
Finished circuit initialization process.
%
```

Figure 11: Output For Pattern 1110

The above fig11 shows the output for pattern (1110). The input value 1110 is applied to the circuit. The t output is produced by input a and b. These two input values are given to OR gate and it wll produce the output at t. The u value is obtained by passing the b value through a NOT gate. The v value is same as the b value. The w output is obtained by passing input b and c through a AND gate. The y output is obtained by passing d value via NOT gate and combinig this with a w value by the use of OR gate. The z output is produced by passing c and via AND gate. The circuit will produce the output as 101110. The circuit will produce the delay as 5.934ns

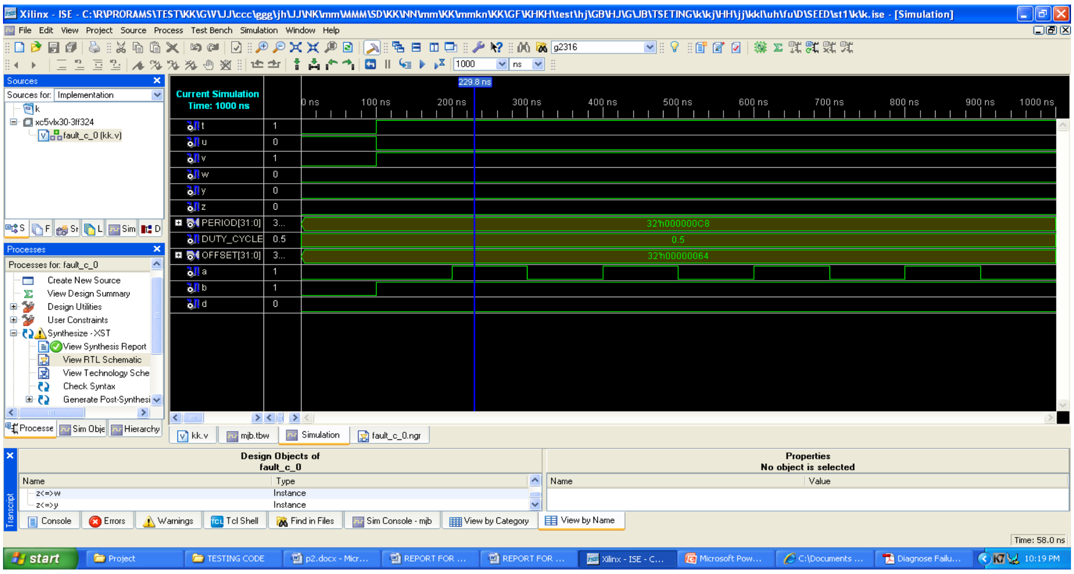


Figure 12: Faulty Output For Pattern 1110

The above fig 12 shows the faulty output for pattern 3 (1110). The input value 1110 is applied to the circuit. Faults are injected in b,q and c. The t output is produced by input a and b. These two input values are given to OR gate and it wll produce the output at t. The u value is obtained by passing the b value through a NOT gate. The v value is same as the b value. The w output is obtained by passing input b and c through a AND gate. The y output is obtained by passing d value via NOT gate and combinig this with a w value by the use of OR gate. The z output is produced by passing c and via AND gate. and it produce the output as 101000 instead of 101110. The circuit will produce the delay as 5.934ns

Conclusions

This project generate 4 bit test pattern using johnson counter and the generated test pattern was applied as an input to the circuit under test to find the faults. This found all the faults in the circuit during testing. And this program run faster compared to the previous method. So it reduces delay that was produced in the circuit. In further analysis I will implement fault element graph which is used to find multiple fault at the same time.

References

[1] B.Suganya, A.Jyothi Test Patterns Generator For Bist Schemes Using Multiple Sic Vectors in 2nd International Conference on Science,

Engineering and Management,Srinivasan Engineering college, TamilNadu, India, March 28-29,2014

[2] James Chien-Mo Li, Member, IEEE, and Edward J. McCluskey, Life Fellow, IEEE "Diagnosis of Resistive-Open and Stuck-Open Defects in Digital CMOS ICs"IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, VOL. 24, NO. 11, NOVEMBER 2005

[3] Leendert M. Huisman "Diagnosing Arbitrary Defects in Logic Designs Using Single Location at a Time (SLAT)" IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, VOL. 23, NO. 1, JANUARY 2004

[4] H. Takahashi, N. Yanagida, and Y. Takamatsu, "Multiple stuck-at fault diagnosis in combinational circuits based on restricted single sensitized paths," in *Proc. 2nd Asian Test Symp.*, Nov. 1993, pp. 185–190.

[5] N. Yanagida, T. Hiroshi, and T. Yuzo, "Efficiency improvements formultiple fault diagnosis of combinational circuits," in Proc. 3rd Asian *Test Symp.*, Nov. 1994, pp. 82–87.

[6] Y.-C. Lin and K.-T. Cheng, "Multiple-fault diagnosis based on singlefault activation and single-output observation," in Proc. Design, Autom. *Test Eur.*, Mar. 2006, pp. 424–429.

[7] Y.-C. Lin, F. Lu, and K.-T. Cheng, "Multiple-fault diagnosis based on adaptive diagnostic test pattern generation," *IEEE Trans. Comput. Aided Design Integr. Circuits Syst.*, vol. 26, no. 5, pp. 932–942, May 2007.

[8] S. Venkataraman and S. Drummonds, "POIROT: A logic fault diagnosis tool and its applications," in *Proc. Int. Test Conf.*, Oct. 2000, pp. 253–262.

[9] S.-Y. Huang, "Toward the logical defect diagnosis for partial-scan designs," in *Proc. Asia South Pacific Design Autom. Conf.*, Feb. 2001, pp. 313–318.

[10] S.-Y. Huang, "On improving the accuracy of multiple defect diagnosis," in *Proc. IEEE VLSI Test Symp.*, Apr. 2001, pp. 34–39.

[11] X. Tang, W.-T. Cheng, R. Guo, and S. M. Reddy, "Diagnosis of multiple physical defects using logic fault models," in *Proc. 19th IEEE Asian Test Symp.*, Dec. 2010, pp. 94–99.

[12] B. Boppana, R. Mukherjee, J. Jain, and M. Fujita, "Multiple error diagnosis based on Xlists," in *Proc. 36th Design Autom. Conf.*, Jun. 1999, pp. 660–665.

[13] X. Wen, T. Miyoshi, S. Kajihara, and L.-T. Wang, "On per-test fault diagnosis using the X-fault model," in *Proc. IEEE/ACM Int. Conf. Comput. Aided Design*, Nov. 2004, pp. 633–640.

[14] T. Bartenstein, D. Heaberlin, L. Huisman, and D. Sliwinski, "Diagnosing combinational logic designs using the single location at-a-time (SLAT) paradigm," in *Proc. Int. Test Conf.*, Nov. 2001, pp. 287–296.

[15] D. Lavo, I. Hartanto, and T. Larrabee, "Multiplets, models, and the search for meaning: Improving per-test fault diagnosis," in *Proc. Int. Test Conf.*, Jan. 2002, pp. 250–259.