Logical Fault Modelling Algorithm for Stuck-at-Fault

K. Mariya Priyadarshini, Kurra Harshitha, Pritika Kanchan, K. Mercy Romitha

Abstract–With miniaturization happening around with the technology, it's very important that the faults associated with these circuitsto get accurate results, especially electronics circuits. Besides, finding these faults is a tough job as there will be several test inputs that needs to be tested to check the circuit is fault free or not. Stuck at line is a deficiency prototype used as a part of computerized testing circuit. When any of the line in the circuit is stuck permanently at power supply or ground giving unwanted output, this is called fault. This paper describes about a technique that can be used to find stuck at fault and display the test vectors that generates the faulty output.Any self-assertive different shortcoming in combinational and consecutive circuits can be mimicked and tried utilizing the displayed stuck at fault model. High fault coverage is especially significant during assembling test, and strategies. Stuck at fault results are presented and detected. The outcomes of single stuck at faults are presented in this paper using Verilog code.

Keywords—Fault Detection, Test Pattern Generation, Stuck- at-fault, Single Stuck-at-fault, Fault Algorithm

I. INTRODUCTION

Fault Detection and diagnosing (FDD) plays a vital role within the safety and responsibleness of commercial method operations [1]. Accordingly, digital systems designed with a lot of extra complexity, the fault testing and identification of digital circuits becomes a vital and indispensable part of the producing method. Besides, manufacturing of electronics products and their miniaturization needs to give accurate results, which is only possible by testing the circuit to find the faults, diagnosing the circuit and correcting the faults by using fault tolerance methods or any other methods. The device quality and reliability will increase. Circuits are shrinking in physical size whereas growing both in speed and range of capabilities. The electronics products like chips or VLSI circuits are tested for defects. But it's impractical to come up with or apply vectors to check all possible defects in a chip. So, defects are modeled as faults to ease the test generation process.

Among the various existing fault models, stuck-at fault model is widely accepted because of its closeness to actual defects and the algorithmic possibilities it offers for generating test vectors [2][3]. There are several problems that occurs in the electronics circuits like races in the circuit, delay faults, etc. [4]. If two or more feedback signals change at the same time, a race may exist in a circuit. The race is vital to the order of changes that may affect the final state of the circuit. Another problem with the design of ASC's is thus to avoid such critical races. An automatic device failure is the unintended difference between the hardware implemented and its planned model. A fault is considered the representation of a fault at an abstract functional level.Deficiency model falls under one of the accompanying presumptions. Single Fault Presumption (Single stuck fault) is a presumption in which just one deficiency can happen in a circuit and Multiple Fault Presumption (Multiple stuck fault), a presumption in which multiple deficiencies can happen. Individual flag and sticks are thought to be stuck at Logical '1', '0' and 'X'. A simple model of stuck-at-fault (SA1 and SA0) is shown in the below given figure 1 and figure 2.

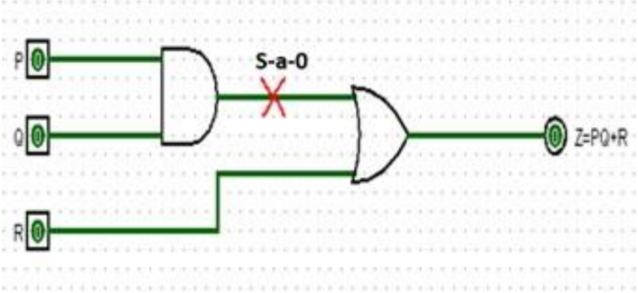


Fig 1: Single Stuck-at-fault circuit

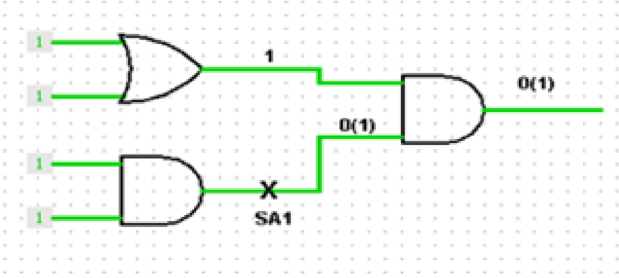


Fig 2: Single Stuck-at-fault (SA1) Circuit

II. PROPOSED SYSTEM MODEL

A. Circuit Models

In this paper, a digital combinational circuit with two stages is being studied and presented as a prototype. In order to make the performance analysis easier, a practical digital circuit of different logic gates like AND gate, OR gate, etc. is chosen [4].

Manuscript received on 03 February 2021 | Revised Manuscript received on 20 February 2021 | Manuscript Accepted on 15 March 2021 | Manuscript published on 30 March 2021.

* Correspondence Author

Ms. K. Mariya Priyadarshini*, Assistant Professor Electronics & Communication Engineering, Guntur, India Email: mariyapriyadarshini@kluniversity.in

Kurra Harshitha, Electronics & Communication Engineering KLEF Guntur, India Email: harshithakurra9@gmail.com

Pritika Kanchan, Electronics & Communication Engineering KLEF Guntur, India Email: pritikakanchan13@gmail.com

K. Mercy Romitha, Electronics & Communication Engineering KLEF Guntur, India Email: mercy.romitha12@gmail.com

In addition, the methods acquired to test these circuits are generally enough to similar circuits that consists of modified logic gates like AND gate with some simple modifications and so on. The delay of these logic gates used is presumed to be identical.

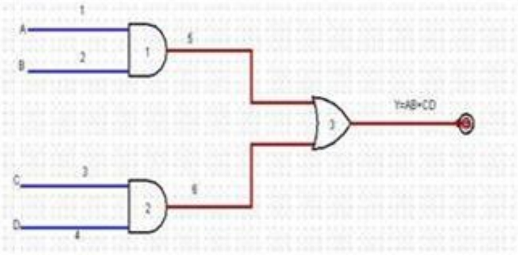


Fig 3: Testing Circuit

In figure 3, the simple testing combinational circuit having two stages is shown. A, B, C, D are the primary inputs and Y = AB + CD is the fault free output of the circuit under test. The gates are numbered and so are the interconnects.

B. Problem Model

The problem of deterministically creating a test example for a deficit (Sa {0,1}) is to detect the assignment of a combination of logical values (0 or 1) to the main inputs that recreate the deficiency (line support) and to monitor the deficiency [5]. The objective is to create a sample model or to consider the flaw as recurring for each fault in the overview of a defect and to construct a test of collection of measuring defects. Further, the stuck – at – fault phenomenon occurs only in the case of hardware; however, it is not possible to create a combinational circuit or any such physical model in software and test the faults, generate the test vectors. The software model will give the output as per the characteristics of used combinational gates or any elements and the given test inputs. The realization of such fault model either stuck-at-one or stuck-at-zero is possible only if some other elements are used in the circuit. In realizing such model in this paper, we have introduced a multiplexer. The selection lines will determine whether the circuit is at single stuck-at-fault,i.e., SA0 or SA1. If the selection line is 0, stuck-at-zero fault will propagate through the circuit and if 1 is given then stuck-at one fault will propagate through the circuit. Besides, for rest of the inputs of selection line, the fault free output will be obtained. By simulating the given circuit in Vivado, the test vectors for the faulty outputs can be realized and counted. The proposed circuit model can also be used for realizing the multiple-stuck-at-fault. The realized model for single stuck-stuck-at fault is presented in the figure 4.

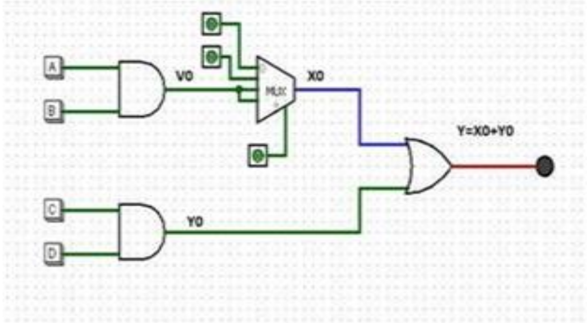


Fig 4 : Proposed 2 – stage combinational circuit with stuck – at – fault model

C. Serial Simulation

This is the most basic and simple algorithm to simulate faults. The circuit is first modelled in true value mode, and all the vectors and primary outputs are saved to a file. Then, on – by – one simulation of all faulty circuits is performed. This is achieved using the true – valued simulator, adjusting the circuit for targeted fault. The performance of faulty circuit increasingly contradicts the saved true values as the simulation goes forward. Once the correlation shows the identification of the desired fault, the simulation of the fault circuit if turned off. It simulates all faults serially. Logic faults influences the status of logic signal. The state can normally be modelled as {0,1, X (unknown), Z (High impedance)} and a fault may turn a true value into another.

III. TYPES OF FAULTS

As all feasible physical defects cannot be enumerated and tests developed, defects are modelled as faults. These models mimic the physical defect behavior while facilitating the test generation process. Logical faults can be designed as structural and functional faults.

The faults acknowledged in this analysis is fixed or permanent or non – transient faults, implying that the fault is perpetual if they are not repaired. Many of the faults in the currently used circuits such as Transistor – Transistor Logic Circuit (TTL), DTL or RTL are those that cause a node to be stuck either at logic ‘1’ (SA1) or logic ‘0’(SA0). It is logically reasonable to limit our attention to just one class of faults as most circuits’ faults have symptomatically the same impact [7]. Multiple Stuck – at – fault can be defined as the simultaneous occurrence of both the stuck – at – fault model, i.e., SA0 and SA1 in the same circuit [3]. This paper discusses only about the single stuck-at-fault. However, it can be extended to multiple faults. A circuit contains other defects other than defects that cannot be observed. If there is a test to determine when or not the circuit has such a fault, a combinational circuit fault is said to be detectable else termed as undetectable [7]. A combinational circuit is termed as irredundant if any logical fault raised at any part of the circuit induces a change in the switching function that a fault free circuit recognizes.

All the SA0 and SA1 faults in the circuit are measurable only if the function realized by the circuit is a minimized one. Two faults are considered as identical if the output function of both these faults is having same truth table. In other terms, we cannot consider a test to differentiate between the two faults based on output information. On the other hand, if there is any such test, applying which on a circuit gives the output response for both the faults, then both the faults are said to be distinguishable.

IV. SINGLE STUCK AT FAULT MODEL

The single stuck at fault (SSF) model is often known as classical or standard fault model as it was widely used and studied. While its reliability is not absolute, the following characteristics is useful [10].

- This represents a lot of physical faults.
- It is technologically independent since the concept of single line stuck at a logical value can be used for any structural model.

- Researches have shown that tests for SSFs' often recognize several non – classical faults.
- The number of single stuck at fault in a circuit is small than any other fault models. In addition, failure collapsing techniques may reduce the number of defects to be directly evaluated.
- For other type of faults, SSFs may be used to design. It can be illustrated from figure 5.

Table 1: Single stuck at fault model table

	X1	X2	X3	X4`	X5
T1	1	0	0	1	0
T2	1	0	0	0	0
T3	0	1	0	0	0
T4	0	1	1	0	0

In the original circuit, a multiplexer (selector) is inserted to model a fault that alters the action of signal line z.
$z' = z \; if \; f = 0$
$z' = z \quad f \; f = 1$
With f = 0, the new circuit functions the same as the original circuit and it can recognize any faulty function zf operation by adding fault f sa1. $x = z$, for example, will create a functional fault that changes the inverter function , i.e., from $z = x \; to \; z = x$.
Similarly, a delaying fault would be generated if x was connected to zf via an inverter with different fault. Even though it is flexible, the non – classical modelling of faults is restricted by the necessity ofsignificantly increasing the size of model.

V. RESULTS

Fault simulation consists of simulating a circuit in presence of fault. The faults identified by a test set T can be found by contrasting the faulty circuit response to the response of circuit that is fault free using the same set T. Fault reproduction includes recreating a circuit's conduct within the sight of deficiencies. Contrasting the broken reaction of the circuit to that of the issue free reaction utilizing a similar test set t, can decide the deficiencies recognized by the same test set itself. Flaw re – enactment has numerous applications, for example, test set assessment, shortcoming focused test age, issue lexicons development, and assessment of circuit activity in the region of inadequacies. The simulation results and other response of the circuit when simulated by using Verilog code in Vivado tool gave the following responses.

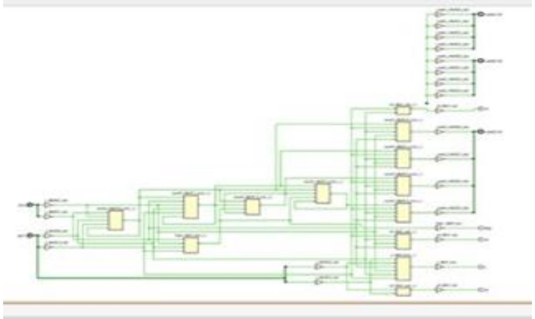


Fig 5: Designing of model using SSF

A. Schematic Diagram

The schematic diagram generated by the Vivado tool from the simulation of Verilog code for circuit in figure 4 in shown below. The schematic diagram consists of 36 cells, 23 I/O ports and 43 nets.

B. Simulation Result

As the circuit has 4 primary inputs, the input will begin from 0000 to 1111. Output obtained through the simulation is presented in figure 6 and figure 7. Further, the result also shows the count of test cases that generates the stuck-at- zero faults for the given 2 – stage combinational circuit and so for the stuck – at – one fault. It also gives idea about which are those test vectors for which special care of the circuit must be takes.

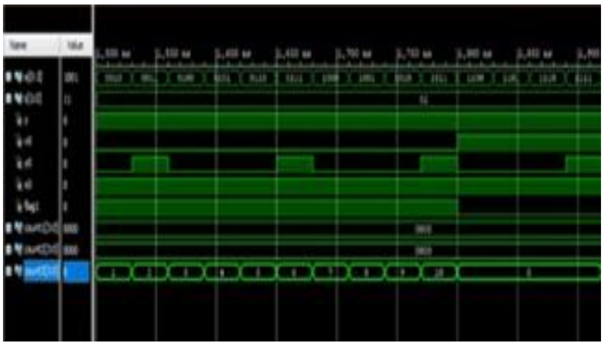


Fig 6: Simulation result for stuck – at – one fault

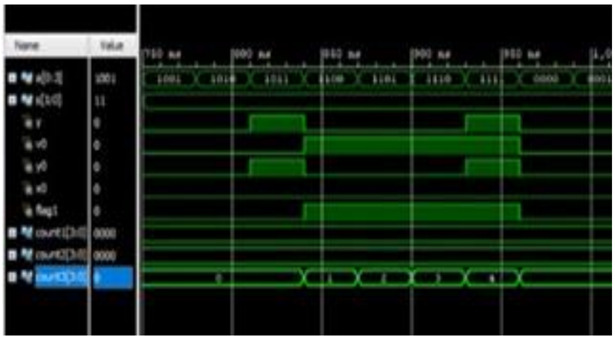
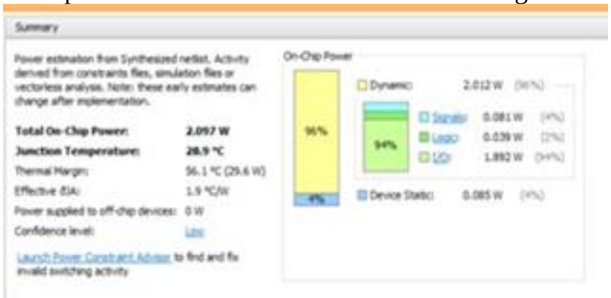


Fig 7: Simulation result for stuck – at – zero fault

C. Power Utilization

The power utilized by the circuit in Vivado is expressed in the figure 9. It represents how the on – chip components like I/O ports and others consume power. This helps in constructing the circuit in a such a way that less power is being consumed by the circuit. It is very necessary in the world of miniaturization. Moreover, the heat produced by the circuit can destroy the other components on the chip. Thus, the temperature estimation can also be found using Vivado.



Summary

Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power: **2.097 W**
Junction Temperature: **28.9 °C**
Thermal Margin: 56.1 °C (29.6 W)
Effective θJA: 1.9 °C/W
Power supplied to off-chip devices: 0 W
Confidence level: Low

Launch Power Constraint Advisor to find and fix invalid switching activity

On-Chip Power

Dynamic:	2.012 W	(96%)
Signals:	0.081 W	(4%)
Logic:	0.039 W	(2%)
I/O:	1.892 W	(94%)
Device Static:	0.085 W	(4%)

Fig 8: Schematic Diagram of circuit in figure 4 using Vivado

VI. UTILIZATION DESIGN INFORMATION

After design implementation, the device utilization can be verified using the Utilization Design Information Report. If the utilization is not expected, various techniques can be implemented in order to the expected result.

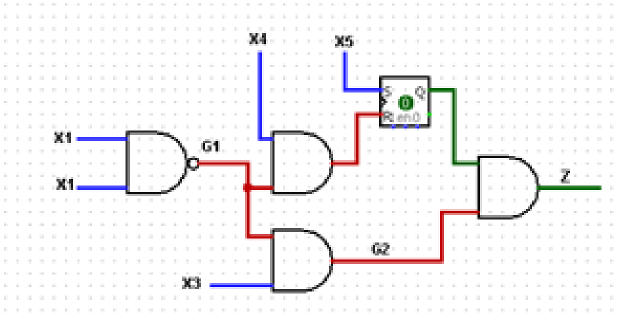
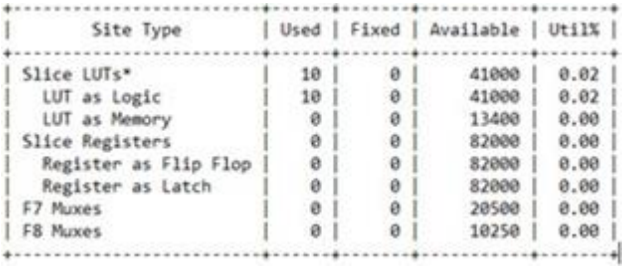


Fig 9: Power consumption and temperature estimation using Vivado tool

A. Slice Logic

The slice logic distribution, the synthesis determines how the model is assembled and inserted into the targeted structure and Map includes information about utilization after packaging and placement.

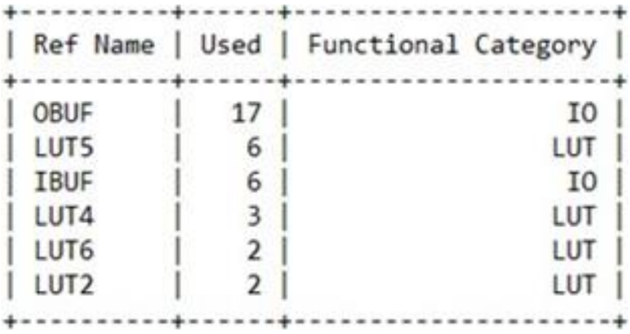


```
+----------------------------+------+-------+-----------+-------+
|          Site Type         | Used | Fixed | Available | Util% |
+----------------------------+------+-------+-----------+-------+
| Slice LUTs*                |   10 |     0 |     41000 |  0.02 |
|   LUT as Logic             |   10 |     0 |     41000 |  0.02 |
|   LUT as Memory            |    0 |     0 |     13400 |  0.00 |
| Slice Registers            |    0 |     0 |     82000 |  0.00 |
|   Register as Flip Flop    |    0 |     0 |     82000 |  0.00 |
|   Register as Latch        |    0 |     0 |     82000 |  0.00 |
| F7 Muxes                   |    0 |     0 |     20500 |  0.00 |
| F8 Muxes                   |    0 |     0 |     10250 |  0.00 |
+----------------------------+------+-------+-----------+-------+
```

Fig 10: Slice Logic Distribution

B. Primitives

They are basic logic blocks used in FPGA. The primitives for the circuit in figure 4, generated using Vivado tool is shown below.



```
+----------+------+---------------------+
| Ref Name | Used | Functional Category |
+----------+------+---------------------+
| OBUF     |   17 |                  IO |
| LUT5     |    6 |                 LUT |
| IBUF     |    6 |                  IO |
| LUT4     |    3 |                 LUT |
| LUT6     |    2 |                 LUT |
| LUT2     |    2 |                 LUT |
+----------+------+---------------------+
```

Fig 11: Table for Primitives

VII. CONCLUSION

The deductive shortcoming test system was proposed for the advanced circuit its usage and tried. In this paper the examination results produced by re-enactment program and count results, it is demonstrated that this program is 100% capacity for shortcoming identification and gives exact outcomes. The test system can be stretched out to alter for some other circuits, having distinctive number of contributions just as yields.

REFERENCES

1. J. P. Roth, "Diagnosis of automata failures: A calculus and a method," IBM J. Res. Develop., vol. 10, no. 4, pp. 278-291, July 1966.
2. Novak, O.Gramatov´ a, E.Ubar, R.´ et al. ―Handbook of Testing Electronic Systems‖. Czech Technical University Publishing House, 2005
3. Kalyana R. Kantipudi,"Minimizing N-Detect Tests for Combinational Circuits", M.SC Thesis,Auburn University, Auburn, Alabama ,May 10, 2007.
4. SM. Thamarai , Dr K.Kuppusamy, Dr T.Meyyappan - Fault Detection and Test Minimization Methods for Combinational Circuits - A Survey, International Journal of Computer Trends and Technology- volume 2 Issue 2 – 2011
5. Eichelberger, E. B.‖Hazard Detection in Combinational and Sequential Switching Circuits‖. IBM Journal of Research and Development, Vol. 9, 1965, No. 2, pp. 90–99.
6. M. Schulz, E. Trischler, and T. Sarfert, SOCRATES. ―A Highly Efficient Automatic Test Pattern Generation System‖, IEEE Transactions on Computer-Aided Design, Vol. 7, pp. 1 26- 1 36, 1 988.
7. Samuel C.Lee, Digital Circuits and logic design, Fourth Edition, Prentice Hall of India.
8. Samuel C.Lee, Digital Circuits and logic design, Fourth Edition, Prentice Hall of India.
9. V. Amar and N. Condulmari, "Diagnosis of large combinational networks," IEEE Trans. Electronic Computers, vol. EC-16, pp. 675-680, October 1967.
10. Miron Abramovici, MelVin A. Breuer, Arthur D. Friedman, Digital Systems Testing and Testable Design, A John Wiley and Sons, Inc., Publication,