Multiple Stuck at fault model Analysis

Submitted by Lakshmi Chimakurthy
Reviewer Malinky Ghosh

Abstract

This paper discusses an algorithm to model any given multiple stuck at fault as a single stuck at fault with the insertion of at most n+3 gates, where n is the multiplicity of the targeted fault. The application of this model in circuit optimization, fault diagnosis and testing of multiply testable faults is discussed with examples. Any arbitrary multiple fault in combinational and sequential circuits can be simulated and tested using the presented multiple fault model.

Introduction

Fault simulation plays a significant role in the testing of digital circuits. In fault simulation we use a fault model which models physical faults that may occur in actual circuits. The most commonly used model is a single stuck at fault model which has two faults per line, stuck-at-1 (s-a-1), stuck–at-0 (s-a-0). The basic assumptions that characterize single stuck-at fault model are[2]:

1. Only one line is faulty
2. The faulty line is permanently set to either 0 or 1.
3. The fault can be at an input or output of a gate.

Unlike a single stuck at fault model, a multiple fault model represents a condition caused by the presence of a group of single faults. Research has shown that a large percentage of multiple faults is covered by the single-fault tests, specific design styles and test generation procedures assure the detection of most of the multiple stuck-at faults if the single faults are detected. However the effects of multiple faults cannot be underestimated in VLSI circuits.

If the number of single stuck at faults is O (n) for a circuit size n, the number of multiple faults is $O(2^{m}{}_{o(n)}C_{m})$ for a fault multitude m. The complexity of multiple fault simulation increases with the increase in the number of faults being handled together. But because of the increasing size of VLSI circuits it is important to analyze the properties and characteristics of the multiple faults in terms of both simulation and test generation. Some of the applications of Multiple fault analysis include:

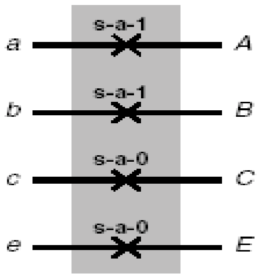
1. improved circuit optimization.
2. improved fault diagnosis
3. better fault coverage

This paper discusses how an arbitrary multiple stuck-at fault in combinational or sequential circuits can be converted in to a logic level single stuck-at fault model which allows effective use of existing tools [1].

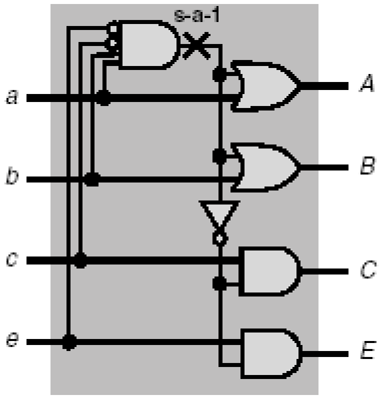
2. Multiple fault model

A multiple stuck at fault of multiplicity n can be modeled as a single stuck at fault by using at most n+3 gates. Consider the circuit in the Figure 1. To convert the multiple stuck at fault in to a single stuck at fault model the following steps have to be performed:

1. A two input gate is inserted in each faulty line. An AND gate is inserted in a line with stuck at 0 fault and an OR gate is inserted in a line with a stuck at 1 fault. These are called In-line gates.
2. The second input of the In-line gates is fed by an n input AND gate. The output of this AND gate feeds the In-line OR gates directly and feeds the In-line AND gates through an inverter. The n inputs to this gate are derived directly from the s-a-1 fault lines and through inversion from s-a-0 lines.



(a) A multiple stuck-at fault.



(b) An equivalent single stuck-at fault.

Figure 1. A Multiple stuck at fault model

Through this conversion when a fault is not activated the output of the In-gate circuit is same as good circuit value. Hence circuit equivalence is preserved. This can be verified using Boolean algebra:

In the modified circuit

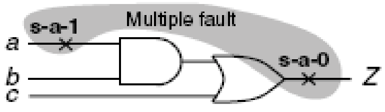
$$A = a + ab\bar{c}\bar{e} = a$$

$$C = c(\overline{ab\bar{c}\bar{e}}) = c(\bar{a} + \bar{b} + c + e)$$

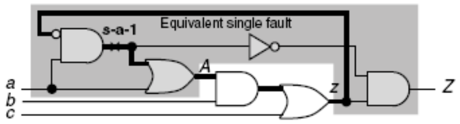
$$= c + c(\bar{a} + \bar{b} + \bar{e}) = c$$

which are same as fault free circuit. Similarly it can be proved for other signal lines also.

The multiple stuck at faults in Figure 1(a) should be same as single stuck at faults in Figure1 (b). By observing the signal values of lines A, B, C, D in both the models it can be verified.



(a) A multiple stuck-at fault.



(b) Equivalent single stuck-at fault with non-functional feedback.

Figure 2. Example of non-functional feedback.

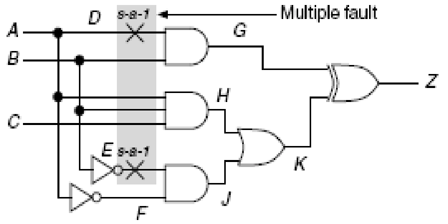
Depending upon the location of fault sites single fault model can produce feedback.

Consider the example shown in Figure 2(a), whose single fault mode is shown in Figure 2(b). A feedback can be observed in the modified circuit but it is never sensitized, irrespective of whether the single fault is present or not the relationship A=a and Z=z are preserved. Hence this feedback is considered as only a structural and not functional. The two circuits are functionally identical.

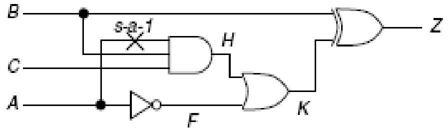
3. Advantages

Multiple stuck at fault analysis can be used to Solve a number of problems. Some of the advantages of this analysis are discussed in this section.

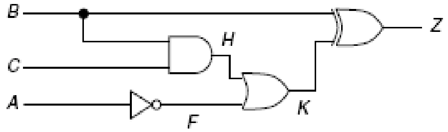
3.1.Circuit Optimization



(a) A single-fault irredundant circuit with a redundant multiple fault.



(b) After removing redundant multiple fault, circuit has a redundant single fault.



(c) Irredundant circuit.

Figure 3. Circuit Optimization

Combinational ATPG programs are used to optimize logic circuits by identifying the redundant single stuck at faults. The optimization process is repeated by considering a single redundant stuck at fault at a time till the circuit becomes totally irredundant. This irredundant circuit can be further optimized by considering multiple faults in many cases.

Consider the example[1] shown in Figure 3(a), all the single stuck at faults are detectable. So the circuit cannot be optimized using single fault model. But we can observe that a multiple fault D s-a-1, E s-a-1 is irredundant. Hence the circuit can be reduced to circuit shown in Figure 3(b). this circuit has a single s-a-1 redundant fault on the input of AND gate coming from signal A. Hence the circuit can be further optimized to circuit in Figure 4(c). In circuits with several redundant single faults groups of these faults can be analyzed by the multiple fault model. Analytical and heuristic procedures can be developed to determine the faults that can be targeted for redundancy analysis.

3.2 . Multiple fault diagnosis

It is important to pin-point the exact fault location to repair the fault in a large circuit. This is called fault diagnosis. Most of the diagnostic procedures are based on single fault tests, the single fault dictionary procedure is very popular. This section explains the improvement in fault diagnosis using multiple fault model.

Consider the circuit in Figure 4[1], a set of six test vectors can detect all the collapsed single stuck at faults. Taking the order of inputs to be A, B, C, D, the test vectors are T_1= 0111, T_2=1110, T_3=1010, T_4=0001, T_5=1101 and T_6=1111 We use a subscript notation to denote a stuck-at fault. For example, A_0 denotes signal A stuck-at-0. In Figure 3, a signal name is either the label on the line or the label of the gate that produces it. A fault dictionary is obtained simulating the vectors without *fault dropping*. Table 1 gives the simulation result in a dictionary format. The outcome of a test is expressed as a binary variable t_i, which is 1 if a fault is detected by T_i or 0, otherwise. Thus, each fault produces a sequence of six binary values, called *test syndrome.* Our *fault dictionary* contains a set of test syndromes associated with all single stuck-at faults. For *no fault*, the fault-free circuit has the test syndrome 000000. Suppose a circuit under test (CUT) contains a multiple fault, ($B1_1$, $C1_1$), i.e., lines B1 and C1 stuck-at-1, simultaneously. The observed test syndrome is shown in Table 1 between two horizontal dividing lines. Since this test syndrome, 001100, does not match with the entries of single stuck-at faults in the dictionary, we use a distance approach.

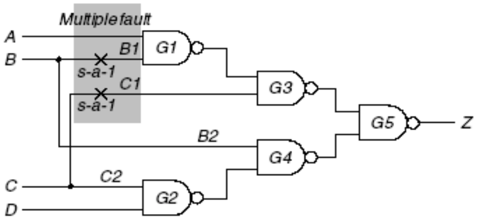
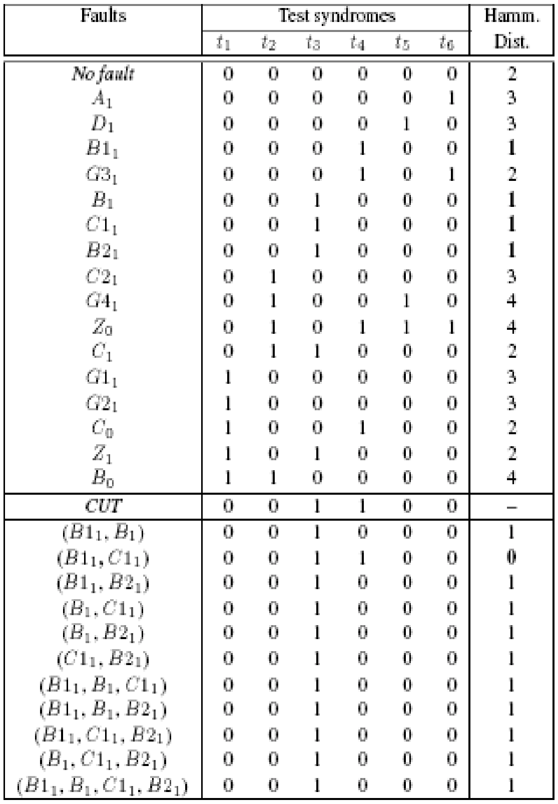


Figure 4. An example circuit for Diagnosis.

The last column, *Hamm. Dist.*, in Table 1 shows the Hamming distance between the observed syndrome and that of each single stuck-at fault.

Table 1[1]. Fault dictionary for diagnosis.



Faults	Test syndromes						Hamm. Dist.
	t_1	t_2	t_3	t_4	t_5	t_6	
No fault	0	0	0	0	0	0	2
A_1	0	0	0	0	0	1	3
D_1	0	0	0	0	1	0	3
$B1_1$	0	0	0	1	0	0	**1**
$G3_1$	0	0	0	1	0	1	2
B_1	0	0	1	0	0	0	**1**
$C1_1$	0	0	1	0	0	0	**1**
$B2_1$	0	0	1	0	0	0	**1**
$C2_1$	0	1	0	0	0	0	3
$G4_1$	0	1	0	0	1	0	4
Z_0	0	1	0	1	1	1	4
C_1	0	1	1	0	0	0	2
$G1_1$	1	0	0	0	0	0	3
$G2_1$	1	0	0	0	0	0	3
C_0	1	0	0	1	0	0	2
Z_1	1	0	1	0	0	0	2
B_0	1	1	0	0	0	0	4
CUT	0	0	1	1	0	0	–
$(B1_1, B_1)$	0	0	1	0	0	0	1
$(B1_1, C1_1)$	0	0	1	1	0	0	**0**
$(B1_1, B2_1)$	0	0	1	0	0	0	1
$(B_1, C1_1)$	0	0	1	0	0	0	1
$(B_1, B2_1)$	0	0	1	0	0	0	1
$(C1_1, B2_1)$	0	0	1	0	0	0	1
$(B1_1, B_1, C1_1)$	0	0	1	0	0	0	1
$(B1_1, B_1, B2_1)$	0	0	1	0	0	0	1
$(B1_1, C1_1, B2_1)$	0	0	1	0	0	0	1
$(B_1, C1_1, B2_1)$	0	0	1	0	0	0	1
$(B1_1, B_1, C1_1, B2_1)$	0	0	1	0	0	0	1

Four faults with the smallest Hamming distance of 1 (shown in boldface) emerge as the suspected fault set, a set of most probable candidates, namely $B1_1$, $C1_1$ $B2_1$, B1. However, we just on the basis of the single-fault tests the suspected fault set cannot be reduced further. Therefore, the multiple-fault model is used to derive the test syndromes assuming any subset of these faults may be present in the CUT. An additional eleven test syndromes, one for each possible grouping are generated by simulating the circuit after injecting the multiple-faults as single stuck-at faults. Observing the Hamming distance in the last column only the test syndrome for the multiple fault pair ($B1_1$, $C1_1$) has a Hamming distance of 0, indicating a perfect match.

In general, the accuracy of diagnosis will heavily depend on the tests used to create fault dictionary. Based on our experiments, we found that the best tests for the multiple fault diagnosis must satisfy following two conditions. First condition is that it must produce a non-zero test syndrome for CUT, differentiate CUT from the fault-free circuit. Second, it should produce a different test syndrome for CUT than the syndromes of single stuck-at faults. Further improvements may be made by generating new tests or using additional tests that target multiple-faults involving the suspected fault set. In other words, the diagnosis may be improved if a longer and/or a different set of test vectors is used.

3.2. Testing untestable Single stuck at faults

Sometimes a group of simultaneously occurring stuck-at faults, might be testable, which might not be testable as a single fault. Consider the example in figure 5 with three redundant single stuck at faults g_1, i_i, j_1.

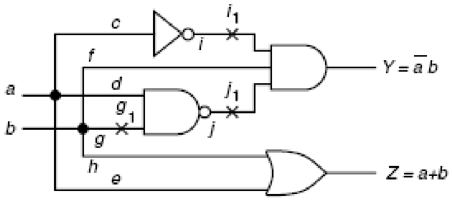


Figure 5. Multiply-testable stuck at faults.

If a typical ATPG program has generated three vectors 00,01 and 10 which can test all the other stuck at faults in the circuit, then by adding adding the vector 11 to the test set we can cover the “singly-untestable faults” when they occur as multiple-faults.

Table 2. Multiple fault tests for the circuit of Figure 5.

Multiple stuck-at fault	Test
(g_1, i_1)	Redundant
(g_1, j_1)	Redundant
(i_1, j_1)	11
(g_1, i_1, j_1)	11

4. Conclusions

The importance of multiple fault analysis and modeling is explained using a new multiple fault model. The fault model studied is not very efficient because it requires a large number of gates. But the model makes efficient use of existing tools to study the characteristics and advantages of multiple fault models as discussed in section 2. The discussion might be beneficial in improving the algorithms for generating tests for multiple stuck at faults.

References

[1] Y. C. Kim, V. D. Agrawal, and K. K. Saluja, “Multiple Faults: Modelling, Simulation and Test,” in *Proc. 15th Int. Conf. on VLSI Design*, Jan. 2002, pp. 592-597.

[2] M. L. Bushnell and V. D. Agarwal, Essentials of Electronic testing for Digital Memory & Mixed signal VLSI-Circuits. Boston: Kluwer Academic Publishers, 2001.

[3] H. Takahashi, K. O. Boateng, “A new method for diagnosing multiple stuck faults using multiple and single fault simulations,” *Proc 17th Int. Conf VLSI Test Symposium, April 1999, pp. 66-72.*