# Detection of Multiple Stuck-at Faults Using Test Pattern of Single Stuck-at Faults

Rahul Gupta (2017 VLSI-10): *Research Scholar, ABV-IIITM, Gwalior*


## 1 ABSTRACT

**THERE are exponentially more multiple faults than single faults in any given circuit design. Only a few additional test patterns are needed to cover all the multiple faults, if the test generation starts from the complete test set for single faults. In this project, we first show the case where test patterns for single faults are sufficient to cover all multiple faults, and then explain in which conditions some of the multiple faults may be overlooked. The method we are using can efficiently generate the complete test set for double faults without traversing all the faults. Since most of the double faults can be detected by the single faults' test set, this method only selects the uncovered double faults by analyzing the propagation paths of single faults and then generating new test patterns only for those uncovered faults.**


## 2 INTRODUCTION

There are techniques that are used for generating test patterns which can efficiently detect all single faults in a circuit but no practical technique exist for detection of simultaneously occurring multiple faults. Previous research works are capable to generate tests for multiple faults. However, these methods does not guarantee to cover all the faults as well as these methods cannot handle largest ISCAS benchmarks.

MSA faults in the large circuit is very difficult to be completely checked due to its huge number of fault combinations. If there are m possible faulty locations in the circuit, then $3^m - 1$ combinations of multiple faults are practically impossible to be detected one by one, when m is large. If m possible SSA faults are there in the circuit, the time complexity to inspect all SSA faults is O(m), while it becomes O($m^2$) for DSA faults, O($m^3$) for triple faults and O($m^n$) for n faults.

Our method first checks the propagation path of an SSA fault to find whether other SSA faults block its fault propagation or not [1]. In this manner, we generate a list of undetected DSA faults. Therefore, instead of traversing the entire DSA fault list, we only deal with a very small subset of DSA faults that are not detectable by SSA test patterns. The SSA fault pairs which mutually mask the propagation paths with each other are classified as the uncovered DSA faults. Therefore, the complexity to find the undetected DSA faults is the same as the SSA fault analysis, that is O(m) if there are m possible SSA faults.


- *Rahul Gupta(2017VLSI-10), ABV-Indian Institute of Information Technology and Management Gwalior, India, 474015.*


## 3 CONCEPT OF MASKING

When the fault is blocked at a gate and cannot be propagated to any output then this gate is called Masking gate. In other words, the fault cannot be observed at any output when the current test pattern is applied.

The term masking gate is used to describe the gate at which, due to MSA fault or SSA fault with multiple propagation paths, more than one input may have faulty values. Depending on the expected values at the gates inputs, the faulty signals may correct each other, thus masking the fault. In other words, the test pattern no longer detects the fault.

For any basic two-input gate (AND/NAND/OR/NOR), in order to mask a MSA fault, the faulty signals which are propagated to its inputs by the respective faults must both be faulty and of opposite polarity. In other words, one input must be D, and the other must be D'.

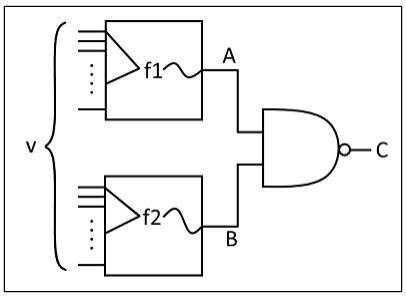



Figure 1. Propagation paths of faults f1 and f2 intersect at a gate

Consider a combinational circuit with primitive gates. We assume all the gates are 2-input NAND gates. Lets assume that there exists a DSA fault, f1, f2, which consists of the two non-equivalent SSA faults, f1 and f2. Assuming there exists a pair of test patterns, v1 and v2, which detect f1 and f2, respectively. Assuming that the circuit is a tree with one root node, it is guaranteed that the propagation path of one fault will intersect with the other. f2 is completely unaffected by f1 regardless of the value of f1 as shown in Fig. 1. Here exists three cases as follows.

1) f2 does not assign the value at wire B, which means that the value at wire B can be changed by assigning

the values of inputs to gate, accordingly v1 can set wire B to 1 to propagate f1 to the output.

2) f2 is equal to stuck-at 1 fault at wire B. Therefore, regardless of the input value at gate, wire B is fixed at 1 due to the fault f2. Then f1 is detected by v1.
3) f2 is equal to stuck-at 0 fault at wire B, and hence the propagation of f1 will be masked when using test pattern v1 to detect it. In this case, the primary output value at wire C becomes 1 regardless of f1, while the correct value at wire C is 0 under the test pattern v2, since the precondition is that f2 can be detected by v2. Therefore, v2 can detect f2 because the faulty value and the correct value at the output wire C are different under the test pattern v2.

Hence the DSA fault f1, f2 can be detected by test pattern v1 in case1 and case2, while it can be detected by test pattern v2 in case3. These are all the cases which can happen when two faults intersect at the same gate. Therefore, the DSA fault f1, f2 is detectable by v1 or v2, when the circuit has no re-convergence.

## 4 DSA FAULT SELECTION

There are three possible cases when DSA faults occur, as follows.

1) f1 and f2 are propagated in two different paths without any intersection, as shown in Fig. 2. The DSA fault can be detected by v1 or v2, since the propagation paths of the two faults are not effected at all because of different propagation paths.
2) When one fault lies in the propagation path of another fault as shown in Fig. 3, f1 is completely unaffected by f2 regardless of the value of f2. Therefore, the DSA fault must be detected by v1, and vice versa.
3) When one fault violates the path constraints of another fault as shown in Fig. 4. Two subcases are discussed below:
    a) If only f1 violates the path constraints of f2 but f2 does not block the propagation path of f1, the DSA faults can be detected by v1.
    b) On the other hand, if two SSA faults mutually violate the path constraints of each other, f1, f2 cannot be detected by either v1 or v2, if both SSA faults have only one propagation path.

## 5 DSA FAULT SET USING TEST PATTERN OF SSA FAULTS

DSA fault which may be undetected due to fanout is shown in Fig. 5 and Table 1.

The test pattern 1001 will detect both stuck-at 1 faults at locations f1 and f2. However, this test pattern will not detect the case when both faults are simultaneously active. Also, the DSA fault is not redundant, as either of the test patterns 1000 or 0001 will detect it.

In other words, this DSA fault could potentially slip through the testing phase undetected.

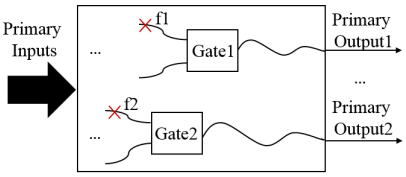



Figure 2. Propagation paths of two faults have no intersection

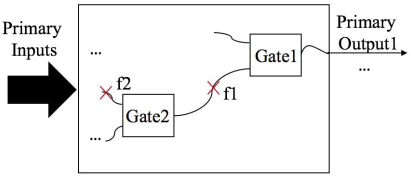



Figure 3. One fault lies in the propagation path of another fault

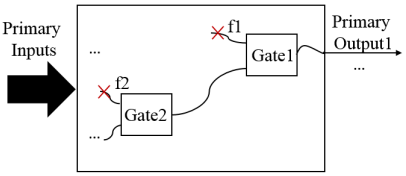



Figure 4. Path constraints of one fault is violated by another fault

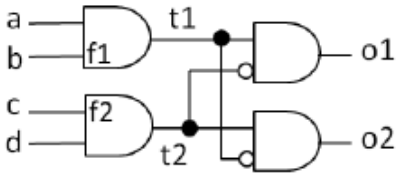



Figure 5. Circuit in which DSA fault may be overlooked, f1 and f2 are stuck-at 1 faults.

## 6 PROPOSED ALGORITHM FOR DOUBLE FAULTS

One of the main issues with using single fault ATPG for the detection of multiple faults is that a second fault may interfere with the propagation of the first one. Therefore, we propose a method similar to the compact pattern generation for SSA faults. In this, if given some DSA fault f1, f2, then as long as f2 does not interfere with the propagation of f1, f1 will be detected, and therefore f1, f2 will also be detected. The method focuses on putting path constraints on the circuit such that a fault is guaranteed to propagate to a primary output. Once a test pattern is found for some

| abcd | Fault | $t_1$ | $t_2$ | $o_1$ | $o_2$ |
|---|---|---|---|---|---|
| 10xx | SSA 1, $f_1$ | D' | 0/1 | D'/0 | 0/D |
| xx01 | SSA 1, $f_2$ | 0/1 | D' | 0/D | D'/0 |
| 1001 | DSA 1, $\{f_1, f_2\}$ | D' | D' | 0 | 0 |
| 1000 | DSA 1, $\{f_1, f_2\}$ | D' | 0 | D' | 0 |
| 0001 | DSA 1, $\{f_1, f_2\}$ | 0 | D' | 0 | D' |

Table 1
Test pattern 1001 detects both SSA faults, but it does not detect the DSA fault.

fault and the constraints are in place, any other fault which may interfere with the constraints can easily be identified.

1) Generate a list of all eligible SSA faults.
2) Pick a focus fault, fi (If possible, give high priority to faults at fanout wires closest to primary outputs and low priority to primary inputs).
3) Generate a test pattern, vi, for fi.
4) Based on fi and vi, find the necessary path constraints for the fault propagation.
5) Find other stuck-at faults which violate the constraints and also mask the focus fault.
6) Add all DSA faults which consist of the focus fault paired with each of the path constraint violating faults to a temporary list of potentially undetected DSA faults.
7) Remove focus fault from list of remaining SSA faults.
8) If there are remaining SSA faults, repeat from step 2, else go to step 9.
9) Go through the temporary list of potentially undetected DSA faults, and keep only the faults which are listed twice, i.e. {fi, fj} and {fj, fi}
10) Faults apart from temporary list are detected by the test pattern of SSA.

## 7 Implementation

- Take c17 benchmark circuit.
- A list of all eligible SSA faults can be obtained by fault collapsing through fault equivalence and fault dominance as shown in figure 6.

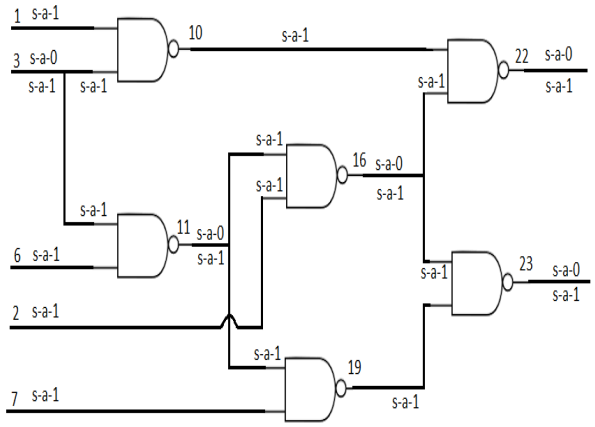



Figure 6. SSA faults in ISCAS c17 benchmark circuit

A list of focus faults fi obtained by ATPG tool Atalanta is shown below.

f1 22 /1
f2 10 /1
f3 22 /0
f4 16→22 /1
f5 3→10 /1
f6 1 /1
f7 3 /0
f8 3 /1
f9 16 /1
f10 16 /0
f11 11→16 /1
f12 2 /1
f13 11 /0
f14 3→11 /1
f15 11 /1
f16 6 /1
f17 23 /1
f18 19 /1
f19 23 /0
f20 16→23 /1
f21 11→19 /1
f22 7 /1

- Now consider focus fault fi=f1= 22/1(stuck at 1 at 22 line).
- Generate a test pattern, vi, for fi.
  PI={1,2,3,6,7}={0,0,x,x,x} and PO={22,23}={0,x}
- Based on fi and vi, the necessary path constraints for the fault propagation is shown in figure 7.

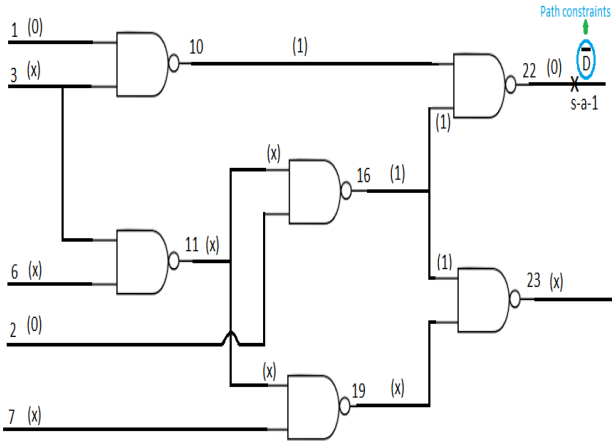



Figure 7. Path constraints for focus fault f1 in ISCAS c17 benchmark circuit

- Now consider those stuck at faults which are activated due to test pattern of focus fault f1.
- Here in this case f6=1/1, f10=16/0 and f12=2/1 faults are activated.
- But f6 and f12 faults are masked before reaching PO.
- Only f10 fault is propagated without violating path constraints of focus fault as shown in figure 8.
- DSA fault sets which are not detected by this test pattern are {f1,f6} and {f1,f12}, shown in figure 9. Thus f6 and f12 are represented by '0'.
- A list of other SSA faults which are not activated by this test pattern are also represented by '0' as it is not detected.
- Only {f1,f10} DSA fault is detected. Thus f10 is represented by '1'.
- Fault masking format for this case along with PI and PO is given by:
  00xxx 0x 10000000010000000000
  It is obtained by using ATPG tool 'Atalanta'.

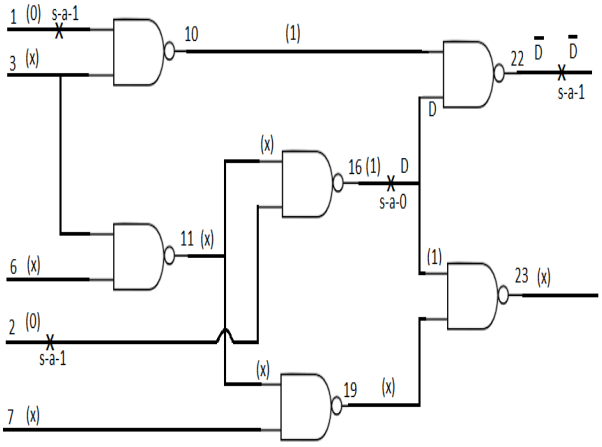
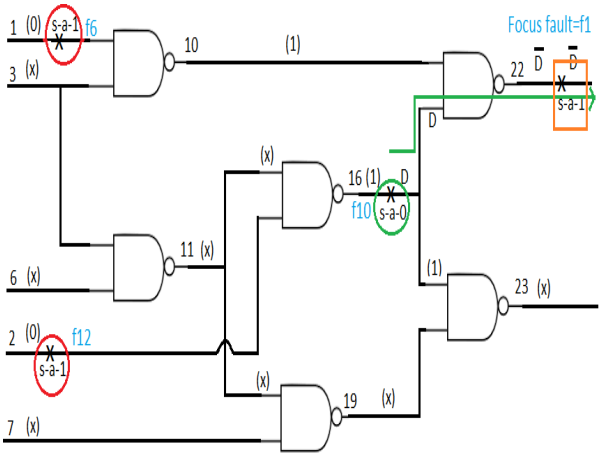



Figure 8. Other activated SSA faults in presence of Focus fault




- From the fault masking format, DSA fault sets which are not detected is given as:
  $\{1,2\},\{1,3\},\{1,4\},\{1,5\},\{1,6\},\{1,7\},\{1,8\},\{1,9\},$
  $\{1,11\},\{1,12\},\{1,13\},\{1,14\},\{1,15\},\{1,16\},$
  $\{1,17\},\{1,18\},\{1,19\},\{1,20\},\{1,21\},\{1,22\}$
- Add all DSA faults which consist of the focus fault paired with each of the path constraint violating faults to a temporary list of potentially undetected DSA faults.
- Potentially undetected DSA fault list is obtained by writing program in C++.
- Now remove focus fault from the list of remaining SSA faults.
- For remaining SSA faults, make them focus fault one by one and repeat the same step that is done for focus fault f1.
- Our program generates potentially undetected DSA fault list for c17 benchmark circuit which consists of 388 DSA faults.
- Now the temporary list of potentially undetected DSA faults is observed and the faults which are listed twice, i.e. {fi, fj} and {fj, fi} are only kept in the list. This list is called undetected fault list which consists of 334 faults. It is not detected by test pattern of SSA faults.
- From this we can obtained the list of DSA faults which is detected by the test pattern of Single stuck-at faults. In c17 benchmark circuit detected faults are 54.

| Benchmark Circuits | Potentially Undetected Faults | Undetected Faults | Detected Faults |
|---|---|---|---|
| c15 | 395 | 348 | 47 |
| c17 | 388 | 334 | 54 |
| c432 | 184536 | 126742 | 57794 |
| s27 | 860 | 756 | 104 |

Table 2
Potentially undetected, undetected and detected DSA faults of ISCAS benchmark circuits

## 8 Result

### 8.1 Analysis of ISCAS benchmarks

The proposed algorithm can be implemented for both combinational and sequential benchmark circuits. For verification we have taken c15, c17 and c432 combinational benchmark circuits and s27 sequential benchmark circuit as shown in Table 2.

## 9 Future Work

Our proposed algorithm still has many areas for improvement, and it is part of our future work. In addition, the proposed approach works well for non-redundant circuits, but we need to consider the method that can deal with the redundant circuits also, since the redundancy often occurs in the circuits.

## References

[1] C. J. Moore, P. Wang, A. M. Gharehbaghi, and M. Fujita, "Test pattern generation for multiple stuck-at faults not covered by test patterns for single faults," in *Circuits and Systems (ISCAS), 2017 IEEE International Symposium on*. IEEE, 2017, pp. 1–4.