# A Fast and Memory-Efficient Diagnostic Fault Simulation for Sequential Circuits

Jer Min Jou and Shung-Chih Chen

Department of Electrical Engineering
National Cheng Kung University
Tainan, Taiwan, R.O.C.

## Abstract

In this paper, a fast and memory-efficient diagnostic fault simulator for sequential circuits is proposed. In it, a two-level optimization technique is developed and used to prompt the processing speed. In the first high level, an efficient list, which stores the indistinguishable faults so far for each fault during simulation, and the list maintaining algorithm are applied, thus the number of diagnostic comparisons is minimized. In the second low level, a bit-parallel comparison is developed to speed up the comparing process. Therefore, the different diagnostic measure reports for a given test set can be generated very quickly. In addition, the simulator is extended to diagnose the single stuck-at device fault. Experimental results show that this diagnostic simulator achieves a significant speedup compared to previous methods.

## 1. Introduction

Three measures, Diagnostic Power (DP), Diagnostic Resolution (DR), and Equivalent Class (EC), for representing the diagnostic capability of a given test set have been proposed for combinational circuits [1],[2]. A diagnostic fault simulator, which is extended from the PROOFS sequential circuit fault simulator [3], proposed another measure, Indistinguishable Class, to provide more diagnostic information for sequential circuits[4]. It used a two-dimensional distinguishability matrix (hereafter, called D-matrix) to record the distinguishability between every pair of faults. Two procedures were proposed to update the D-matrix after each test pattern is simulated. Finally, the indistinguishable classes were reported based on the D-matrix. The time complexity of both procedures is $O(n^2)$, where n is the number of faults. Also, the memory complexity is $O(n^2)$.

In this paper, we propose a two-level optimization technique to speed up the diagnostic process. The first high level minimizes the number of distinguishability comparisons among pairs of faults. The second low level speeds up the comparison process. Experimental results show that this diagnostic simulator achieves a significant speedup compared to the existed ones. The results also show that the time and memory complexities of the proposed method approximate to O(n). The simulator can also diagnose the single stuck-at failure correctly. Due to the space limited, we omit it in this paper.

## 2. Diagnostic measures for sequential circuits

For sequential circuits, since the initial value for each flip-flop is unknown (X), a good or a faulty circuit can produce a value of 0, 1, or X on any primary output (PO). Due to the uncertainty of X, values 0 and 1 are thought as indistinguishable from value X. Hence, equivalent class cannot be applied to sequential circuits directly. Rudnick et al. [4] reported a measure, indistinguishability class, to express the diagnostic capability of a test set for sequential circuits. An indistinguishable class is a set of faults such that every pair of faults in the class is indistinguishable. Two faults f and g are indistinguishable with respect to a test set, if for every PO, either the binary values for f and g are equal, or one of the two values is unknown. Note that the indistinguishability classes may not be disjoint since a fault may appear in different indistinguishability classes due to the uncertainty of the unknown value X.

When the sizes of indistinguishability classes for a given test set are found, the diagnostic capabilities, DP and DR, then can be computed. The number of fully distinguishable faults is the number of indistinguishability classes with size 1; The number of distinguishable fault pairs can be computed from the number of indistinguishable fault pairs since DR is equal to

$$DR = 1 - \frac{\textit{number of indistinguishable fault pairs}}{\textit{total number of fault pairs}}$$

## 3. Previous work

Rudnick et al. [4] used an n-by-n D-matrix to represent the diagnostic information of a given test set, where n is the number of faults. If two faults are distinguishable, the corresponding entries of the D-matrix are set to 1. Two methods were proposed to update the D-matrix.

The first method was to use a two-dimensional array to store all the PO values for each faulty circuit after the simulation of a test pattern. Then each faulty circuit is compared to 32 other faulty circuits at a time. Therefore, the number of comparisons for each test pattern was $(n * m * \lceil n/32 \rceil)$, where m is the number of POs.

The second method was to use two buckets, 1- and 0-bucket, to store the faulty circuits with value one and zero, respectively, at each PO. Since an unknown value is assumed to be indistinguishable from value 0 nor value 1, the 0- and 1-bucket was empty when a PO had an unknown value. Every fault in a 0-bucket distinguished from every fault in the corresponding 1-bucket.

©1994 ACM 0-89791-690-5/94/0011/0723 $3.50

Consequently, for each test pattern, there were $\sum_{i=1}^{m}(n_{i0} * n_{i1})$ entries set, where $n_{i0}$ and $n_{i1}$ are the numbers of faults in the 0- and 1-bucket, respectively, of the i-th PO.

The above methods have two problems. The first one is that the memory requirements are quite huge. These include an n-by-n D-matrix and either the memory for storing the values of all faulty circuits or the 0- and 1-bucket at every PO. The memory complexity is $O(n^2)$. The second one is that many operations are not necessary. For the first method, each faulty circuit compared with all other faulty circuits. In fact, it needs only to compare the faulty circuits that are indistinguishable from it so far. For the second method, all the corresponding entries for the fault pairs distinguished with respect to the current test pattern are set. Actually, some of these entries have been set previously and it is a waste of time to set these entries again. In the following section, we propose our method which can eliminate the drawbacks described above.

## 4. Proposed diagnostic method

To differentiate a pair of faulty circuits under the simulation of a test pattern, their faulty responses must be recorded. This is done by a fault simulator. Several efficient parallel algorithms have been developed to speed up the fault simulation [3],[5]. We ourselves develop a fault simulator, which adopts some concepts proposed in HOPE [5], to obtain the PO values of each faulty circuit.

In the followings, we present a two-level optimization technique to speed up the diagnostic process. Applying the high level technique minimizes the number of comparisons and the low level technique speeds up the comparison process. We describe the two level techniques separately.

### 4.1. Minimization of the number of comparisons

If each faulty circuit only compares with the faulty circuits that are indistinguishable from it so far, then the the number of comparisons is minimized. Based on this, each fault has a list to record the faults that are indistinguishable from it so far. We call such a list an indistinguishable fault list (IFL). At the start of fault diagnosis, every fault is indistinguishable from all other faults. However, we do not construct IFLs for the undetected faults until they are detected, i.e., the IFL for each undetected fault is empty. This is because that we also want to minimize the sizes of memory used. In this way, each fault has a detection flag to indicate the detection of it. Once an undetected fault becomes detected, its detection flag is set and the faults indistinguishable from it are put into its IFL, After each subsequent test pattern is simulated, each detected fault needs only to compare the faulty circuits in its IFL, instead of all other faulty circuits. After the comparison, the fault(s) which is (are) newly distinguished from it with respect to the current test pattern is (are) removed from its IFL. Consequently, the length of the IFL of each detected fault is getting shorter and the number of comparisons is getting fewer. When the IFL of a detected fault becomes empty, the fault is said to be fully distinguishable by the given test set and it needn't compare with any other faulty circuits for the subsequent test patterns.

For each test pattern, there may be newly detected fault(s), which is (are) distinguishable from the good circuit response of the current test pattern. Two kinds of faults are put into the IFL of each newly detected fault. The first one includes the detected faults that are indistinguishable from it so far. This is done by traversing the IFL of each detected fault to check whether the newly detected fault is in the IFL or not. If yes, then the detected fault is put into its IFL. The second one includes the undetected faults which are indistinguishable from the newly detected fault with respect to the current test pattern. This is done by comparing its circuit response with the one of all other undetected faulty circuits.

During constructing the IFL for a newly detected fault, not all the faults that are indistinguishable from it are inserted into its IFL. This is because that the distinguishability relation is symmetrical. That is, if a faulty circuit A is distinguishable (indistinguishable) from a faulty circuit B, then the faulty circuit B is distinguishable (indistinguishable) from the faulty circuit A. We need only to compare the PO values of every pair of faulty circuits once instead of twice. Before simulation, each fault is numbered a different number, called fault number. With the symmetrical property, the IFL for a newly detected fault contains 1) the detected faults which are indistinguishable from it and whose fault numbers are greater than the faulty number of it and 2) the undetected faults that are indistinguishable from it.

Based on the above discussions, the algorithm for maintaining the IFL of each detected and newly detected fault for each pattern is listed in Fig. 1. We will describe the *distinguishable* function in the next sub-section.

```
For each previous detected faulty circuit f
  For each faulty circuit g in the $IFL_f$
    If ( distinguishable( f, g ) )
      Remove g from the $IFL_f$
For each undetected fault h
  If ( distinguishable( h, G ) )  /* G is the good circuit */
    For each detected fault f    /* h is newly detected */
      If (h is in the $IFL_f$ and the fault number of h is less
          than that of f )
        Insert f into the $IFL_h$ and remove h from the $IFL_f$
    For each undetected fault g
      If ( ! distinguishable( h , g ) )
        Insert g into the $IFL_h$
    Move h to the detected fault set.
```

Fig. 1. Algorithm for maintaining the IFLs.

After all the test patterns are simulated, the IFL for each undetected fault is constructed. The faults in each IFL include the detected faults, which are indistinguishable from it, and all other undetected faults. For some detected faults, since we applied the symmetrical relation to the diagnostic process, the faults which are indistinguishable from them and with smaller fault numbers to them are put into their IFLs. Then the sets of indistinguishable faults can be obtained by finding all cliques of maximal sizes in the graph created by the IFLs of all the faults. Having the indistinguishability classes, the diagnostic capabilities, DP and DR, are then computed.

## 4.2. Bit-parallel comparison

The *distinguishable* function is a boolean function to determine whether two faulty circuits are distinguishable or not. In order to assess the distinguishability among so many pairs of faults, a bit-parallel method is developed. Two faulty circuits are said to be distinguishable with respect to the current test pattern if any one of their PO values is distinguishable. We first describe how to determine the distinguishability of a PO value of two faulty circuits, and then extend it to determine the distinguishability of 32 PO values.

A two-bit coding technique [4] is used to represent the circuit values 0, 1, and X as (1,0), (0,1), and (0,0), respectively. Let $V(f_i)$ be the value of the i-th PO of the fault f after simulation, and $(V0_{f_i}, V1_{f_i})$ be its coded values, then the distinguishability of a PO value for a pair of faults f and g is defined as:

$$BD(V(f_i), V(g_i)) = (V0_{f_i} \bullet V1_{g_i}) \oplus (V1_{f_i} \bullet V0_{g_i})$$

where $\bullet$ and $\oplus$ are the AND and XOR bit operations, respectively. If the computed value of $BD(V(f_i), V(g_i))$ is equal to 1, faults f and g are distinguishable with respect to the i-th PO. Otherwise, they are indistinguishable.

The above equation is also suitable for determining the distinguishability of a group of 32 PO values, except that the symbol i now represents the i-th group of POs. For clarity, we define the distinguishability of the j-th group of PO values of faults f and g as:

$$WD(V(f_j), V(g_j)) = (V0_{f_j} \bullet V1_{g_j}) \oplus (V1_{f_j} \bullet V0_{g_j})$$

Then we have the *distinguishable* function listed in Fig. 2.

```
Distinguishable(f, g)
{   j = 0;
    while( $j < \lceil m/32 \rceil$ )
    {   if( $WD(V(f_j), V(g_j))$ != 0 )
            return( 1 );
        j = j + 1;
    }
    return( 0 );
}
```

Fig. 2. The *distinguishable* function.

## 5. Experimental results

The proposed sequential diagnostic simulator has been coded in C language and run on a SUN SPARC II workstation with 32 Mbytes memory. Several ISCAS89 sequential benchmark circuits [6] are tested to examine its efficiency. The test sets were generated by the GENTEST sequential circuit test generator [7].

The diagnostic results are compared with the ones shown in [4] and summarized in Table 1. Note that the results of [4] were obtained from a SUN SPARC station IPC with 24 Mbytes memory and the test patterns were generated by the sequential circuit test generator STG3 [8]. For the s35932 circuit, only one-fifth of the original faults was simulated in both simulators.

Table 1. Comparison results.

| circuit | #test | proposed method | | | Rel. [4] | | | speed-up |
|---|---|---|---|---|---|---|---|---|
| | | time | DR | DP | time | DR | DP | |
| s298 | 162 | 1.6s | 88.5 | 0.0 | 9.4s | 94.1 | 0.0 | 5.88 |
| s344 | 91 | 1.5s | 92.4 | 0.0 | 9.1s | 96.7 | 0.0 | 6.07 |
| s400 | 1282 | 24.5s | 85.1 | 0.0 | 121.2s | 82.3 | 0.0 | 4.95 |
| s420 | 173 | 0.3s | 8.1 | 0.0 | 8.5s | 9.5 | 0.0 | 28.33 |
| s526 | 754 | 21.2s | 83.6 | 0.0 | 100.2s | 89.5 | 0.0 | 4.73 |
| s641 | 133 | 0.6s | 97.6 | 39.0 | 29.6s | 97.6 | 39.0 | 49.33 |
| s713 | 107 | 0.9s | 95.8 | 31.0 | 33.4s | 95.8 | 31.0 | 37.11 |
| s820 | 411 | 4.0s | 96.2 | 8.7 | 193.8s | 96.3 | 8.7 | 48.45 |
| s832 | *379 | 3.6s | 96.3 | 8.5 | 190.2s | 96.2 | 8.5 | 52.83 |
| s953 | 16 | 0.2s | 14.9 | 1.4 | 5.6s | 14.9 | 1.3 | 28.00 |
| s1238 | 349 | 1.9s | 99.7 | 81.1 | 196.2s | 99.7 | 81.2 | 103.26 |
| s1423 | 36 | 1.7s | 42.5 | 1.8 | 16.4s | 42.5 | 1.7 | 9.65 |
| s1488 | 590 | 19.5s | 98.8 | 4.0 | 904.2s | 99.1 | 3.9 | 46.37 |
| s1494 | 469 | 24.7s | 97.8 | 3.8 | 741.0s | 98.6 | 3.7 | 30.00 |
| s5378 | 408 | 40.5s | 93.1 | 25.6 | 1.6h | 93.1 | 25.6 | 138.79 |
| s35932 | 86 | 157.7s | 98.6 | 62.6 | 12.5h | 99.3 | 0.0 | 286.04 |

* : 377 tests for Ref. [4]

The time shown in the Table is the time spent on finding the distinguishability among pairs of faults, which is the time spent on updating the D-matrix [4]. Two different methods for updating the D-matrix were proposed and the CPU time of the faster one of each circuit was selected for comparison. Results reveal that our method gets significant improvement compared to the methods of [4]. The speedup factor ranged from 4.75 to 286. For the circuits s5378 and s35932, the faster method of [4] spent 1.56 and 12.53 hours, respectively, but our method only spent 40.5 and 157.7 seconds, respectively.

From Table 1 we can also find that the time spent on finding the indistinguishable faults for each circuit does not increase so fast as the one on updating the D-matrix. It is difficult to compute the time complexity of our diagnostic method due to the dynamic properties of the simulated circuit structure and the list structure. However, we have selected some benchmark circuits with different fault sizes to compute their average execution time and memory usage. They are s298, s526, s832, s1488, and s5378. Results show that both the time and memory complexities of the proposed method approximate to O(n) under the given test sets for these selected circuits.

The diagnostic resolutions and diagnostic powers of both test sets are also shown in Table 1. No significant differences between the two test sets are found. The results of the indistinguishability classes are shown in Table 2. The indistinguishability class with maximum size of each circuit contains the faults undetected by the test set.

## 6. Conclusion

We proposed a very fast and memory-efficient diagnostic fault simulator for sequential circuits in this paper. A two-level optimization technique is developed not only to minimize the number of comparisons for finding the distinguishability between every pair of faulty circuits but also to speed up the comparison process. Several ISCAS89 sequential benchmark circuits were run to test the performance of the proposed diagnostic method. Experimental results showed that the diagnostic method achieved a significant speedup compared to previous methods. Experimental results also showed that the time and memory complexities of our method approximated to O(n). In addition to reporting the diagnostic power and diagnostic resolution for a circuit with respect to a given test set, the simulator was extended to diagnose the single stuck-at device failure correctly.

## References

[1] P. Camurati, D. Medine, P. Prinetto, and M. S. Reorda, "A Diagnostic Test Pattern Generation Algorithm," *Proc. International Test Conference*, pp. 52-58, 1990.

[2] K. Kubiak, S. Parkes, W. K. Fuchs, and R. Saleh, "Exact Evaluation of Diagnostic Test Resolution," *Proc. 29th Design Automation Conference*, pp. 347-352, 1992.

[3] T. M. Niermann, W. -T. Cheng, and J. H. Patel, "PROOFS: A Fast, Memory-Efficient Sequential Circuit Fault Simulator," *IEEE Trans. CAD*, pp. 198-207, Feb. 1992.

[4] E. M. Rudnick, W. K. Fuchs, and J. H. Patel, "Diagnostic Fault Simulation of Sequential Circuits," *Proc. International Test Conference*, pp. 178-186, 1992.

[5] H. K. Lee and D. S. Ha, "HOPE: An Efficient Parallel Fault Simulator for Synchronous Sequential Circuits," *29th Design Automation Conference*, pp. 336-340, 1992.

[6] F. Brglez, D. Bryan, and K. Kozminski, "Combinational Profiles of Sequential Benchmark Circuits," *International Symposium of Circuits & Systems*, pp. 1929-1934, 1989.

[7] W. -T. Cheng and T. J. Chakraborty, "GENTEST, An Automatic Test Generation System for Sequential Circuits," *Computer*, pp. 43-49, Apr. 1989.

[8] W. -T. Cheng and S. Davidson, "Sequential Circuit Test Generator (STG) Benchmark Results," *International Symposium of Circuits & Systems*, pp. 1938-1941, 1989.

Table 2. Sizes of indistinguishable classes.

| circuit | *number of indistinguishability classes by size* | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | > 10 |
| s298 | 3 | 9 | 7 | 15 | '32 | 13 | 9 | 31 | 23 | 6(11),4(12),1(13),1(14),2(15), 1(18),1(44) |
| s344 | 17 | 6 | | 13 | 95 | 88 | 3 | 3 | 10 | 4(11),1(12),1(13) |
| s400 | 25 | 18 | 9 | | | 3 | 20 | 9 | 8 | 28(11),56(12),11(13),5(14), 2(15),4(16),2(20),1(59),1(73) |
| s420 | 2 | | | 1 | | 2 | 1 | 1 | 1 | 1(11),1(12),3(13),2(15),2(16), 1(17),1(20),1(24),1(25),1(432) |
| s526 | 6 | 14 | 4 | 5 | 3 | 2 | 15 | 19 | 18 | 6(11),9(12),25(13),24(14), 19(15),12(16),5(17),3(19),1(20), 1(23),1(25),1(52),1(65),1(137) |
| s641 | 34 | 80 | 17 | 2 | 5 | 2 | 3 | | | 1(64) |
| s713 | 27 | 67 | 20 | 2 | 4 | 6 | 6 | | 6 | 1(111) |
| s820 | 63 | 389 | 51 | 10 | 1 | 2 | | 1 | | 1(14),1(154) |
| s832 | 463 | 51 | 7 | 2 | 2 | | 2 | | | 1(14),1(162) |
| s953 | 13 | 5 | 2 | 2 | 2 | | | 1 | 2 | 1(13),1(14),1(16),1(995) |
| s1238 | 81 | 6 | 1 | | | | | | | 1(72) |
| s1423 | 23 | 11 | 9 | 7 | 5 | 3 | 2 | 3 | 2 | 2(11),2(12),2(14),1(15),1(16), 1(17),1(18),1(22),1(23),1(34), 1(36),1(55),1(1145) |
| s1488 | 109 | 90 | 943 | 58 | 10 | 3 | | | | 1(110) |
| s1494 | 92 | 36 | 19 | 76 | 907 | 62 | 10 | 2 | 1 | 1(11),1(134) |
| s5378 | 534 | 319 | 122 | 91 | 43 | 31 | 15 | 15 | 10 | 10(11),5(12),3(13),4(14), 2(15),1(17),2(23),1(25),1(27), 1(50),1(61),1(71),1(1196) |
| s35932 | 1350 | 138 | 9 | 1 | | | | | | 1(939) |