On Potential Fault Detection in Sequential Circuits

Elizabeth M. Rudnick Janak H. Patel

Center for Reliable &
High-Performance Computing
University of Illinois, Urbana, IL

Irith Pomeranz

Electrical and Computer
Engineering Department
University of Iowa, Iowa City, IA

Abstract

During fault simulation, an approximation frequently used in practice is to declare a fault to be detected after it has been potentially detected a predetermined number of times. This approximation may lead to declaring a fault detected when in fact the fault will not be detected during a standard test application process. We propose an alternative measure of fault detection for potentially detected faults, that is easy to compute, yet its accuracy is significantly higher than the measure based on the number of times a fault is potentially detected. Experimental results are shown to support the accuracy of the new measure.

I Introduction

Fault simulation is used extensively in the test generation process to identify the faults detected by a given test. For efficient processing, fault simulators [1, 2, 3, 4] use a 3-value model in which lines may take on logic 1, logic 0, or unknown (X) values. Sequential circuits are assumed to start in an unknown state, and therefore, X values are initially assigned to all the state lines. Good design practice requires that sequential circuits be initializable, either using hardware reset or synchronizing sequences. A circuit that is initializable under 3-value simulation is *logically initializable*, while a circuit that is initializable considering all possible individual initial states is *functionally initializable* [5, 6, 7]. Faults modify the circuit structure, and certain faults cause the circuit to remain in an unknown state throughout the simulation. For example, the stuck-at-0 fault on the reset line of the flip-flop shown in Figure 1 prevents the flip-flop from being set to a known value, and the faulty circuit remains in an unknown state. Even if the fault effects propagate to a primary output, the value propagated by 3-value simulation will be a known 1 or 0 for the good (fault-free) circuit, but an unknown for the faulty circuit. The fault is thus declared to be *potentially detected* [8].

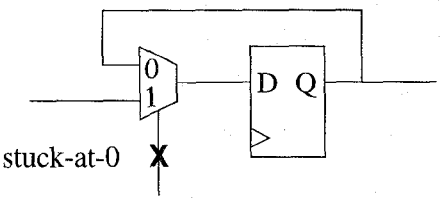

Figure 1: Fault on flip-flop reset line

During testing of the actual physical device, the state lines will always take on 0 or 1 logic values. If a fault is potentially detected for several test vectors, it is likely that either a 1/0 value pair for the good and faulty circuits or a 0/1 value pair will propagate to the primary outputs. In this case, the fault will be detected. Based on this observation, some fault simulators make the approximation that a fault is detected if it is potentially detected for multiple vectors. Options are provided to allow faults to be declared detected after some number n of potential detections; a suggested value for n is 5. This approximation is good for several fault classes. For example, any faults on clock lines can only be potentially detected, since the flip-flop values remain unknown in the faulty circuit. However, they are very likely to be detected in a physical setting, since a constant flip-flop value caused by a clock line fault is likely to cause an incorrect output value. Another such class of faults is stuck-at faults on enable pins of tristate devices. A new fault model has been proposed to handle such faults [9]. Since these classes of faults cannot be resolved in a conventional stuck-at fault simulator even if longer test sequences are considered, it makes sense to remove them early in the fault simulation process to improve the simulation speed. For potentially detected faults outside of the above two classes, the accuracy of the measure of detection based on the number of potential detections is not readily apparent.

*This research was supported in part by DARPA under Contract DABT63-95-C-0069, in part by the Semiconductor Research Corporation under Contract SRC 95-DP-109, by Hewlett-Packard under an equipment grant, and in part by NSF Grant No. MIP-9357581.

0-7803-3540-6/96 $5.00

In this paper, we evaluate the accuracy of the old measure based on the number of potential detections. We will show that a fault may be potentially detected for a number of vectors, but never detected during the test application process, or detected with very low probability. We propose an alternative measure called the *fault detection probability*, to replace the number of potential detections. The fault detection probability is defined as the probability that a fault is detected, given that a machine may start in any state. We estimate this probability by assuming that all states are equally likely. The *estimated fault detection probability*, described in Section II, is easy to compute, yet its accuracy is significantly higher than the accuracy of the number of potential detections. A tool for estimating the detection probability for potentially detected faults is described in Section III. In Section IV, we describe some tools developed for verifying the results of the detection probability estimation tool. Results for several ISCAS89 circuits are presented in Section V, and concluding remarks are given in Section VI.

II Estimated Fault Detection Probability

For a fault to be detected during the application of a test sequence T, the faulty circuit must start from an initial state that produces an output response that is different from any response that the good circuit can produce [10]. To comply with the conventional test application process, we use a single response of the good circuit to T. This response is computed using 3-value simulation starting from the all-unspecified initial state. To imitate the situation that occurs during test application, where the circuit under test starts from a fully-specified (yet unknown) state, we use for the faulty circuit a random sample of 100 fully-specified initial states. In practice, the circuit under test may start from a legal or illegal state; therefore, no restrictions are placed on the 100 initial states. We obtain the faulty circuit response to T starting from each one of the states, and count the number of states for which the fault is detected (i.e., the number of states for which the response of the faulty circuit is different from the response of the good circuit). We refer to the fraction of a random sample of initial states for which a fault is detected as the *estimated fault detection probability*. For a provably detected fault, the estimated fault detection probability is 100%. However, an estimated fault detection probability of 100% does not guarantee that the fault will be detected, since not all the initial states are considered in computing the estimated fault detection probability. Computation of the estimated fault detection probability assumes that each individual initial state occurs with equal probability. While this assumption may not hold, a high detection probability will not be affected by minor variations in the initial state probability. Instead of relying on a limited sample, if we were to include all states in computing the probability, we would get a better estimate, called the *fault detection coefficient* in [10].

Simulations for ISCAS89 circuit s1423 starting from 100 random fully-specified initial states for the faulty circuits show that faults which are potentially detected multiple times are actually detected for less than 50% of the initial states. Thus, these faults may not be detected during test application. In this circuit and other circuits used in this paper, potentially detected faults do not include any clock line faults or any tristate enable faults.

Figure 2 shows a scatter plot of the fault detection probability estimated using a sample of 100 random initial states vs. the number of potential detections for each potentially detected fault in circuit s1423. A set of 89 vectors generated by the HITEC test generator [11] was used. Potential detections are counted at all the primary outputs to which the fault effects propagate.

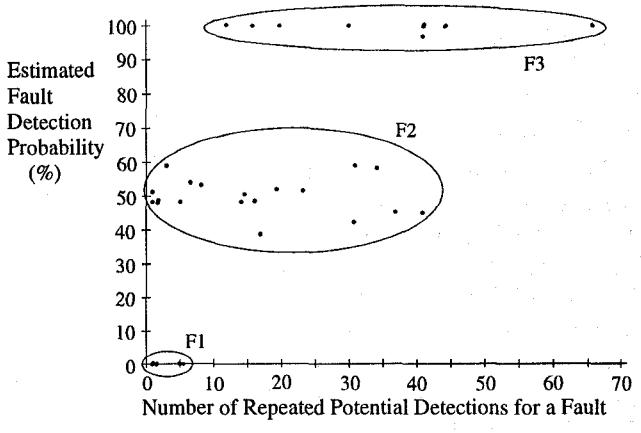

Figure 2: Estimated fault detection probability vs. number of potential detections: s1423

Nine faults (marked F1 in Figure 2) were each potentially detected between one and five times starting from an all-unknown state, but were not detected for any of the 100 random fully-specified initial states. These faults may be erroneously considered to be detected by the measure based on the number of potential detections. Nineteen faults (marked F2 in Figure 2) fall into the middle range and were detected for between 38% and 59% of the initial states. The corresponding number of potential detections ranged from 1 to 41. Ten faults (marked F3 in Figure 2) were detected for between 97% and 100% of the initial states, and the

number of potential detections for these faults ranged from 12 to 66. As illustrated in the figure, declaring a fault to be detected after n potential detections is inaccurate for this circuit. If a value of 5 is used for n, faults that are not detected for any starting state or that are detected for only half of the possible starting states are declared to be detected. If a value of 10, 20, 30, or 40 is used for n, faults that are not detected for half of the possible starting states are declared to be detected. In addition, faults that are detected for 100% of the possible starting states (and thus may be provably detected or detected with high probability) are not identified as detected if a value of 20, 30, or 40 is used for n.

Results for circuits s382 and s5378 further illustrate the inaccuracy of the measure based on the number of potential detections, as shown in Figures 3 and 4. Fault detection probabilities were estimated using samples of 100 random initial states. For s382, two faults (marked

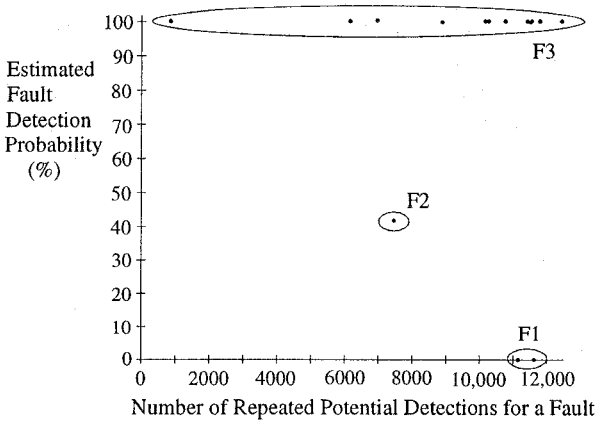

Figure 3: Estimated fault detection probability vs. number of potential detections: s382

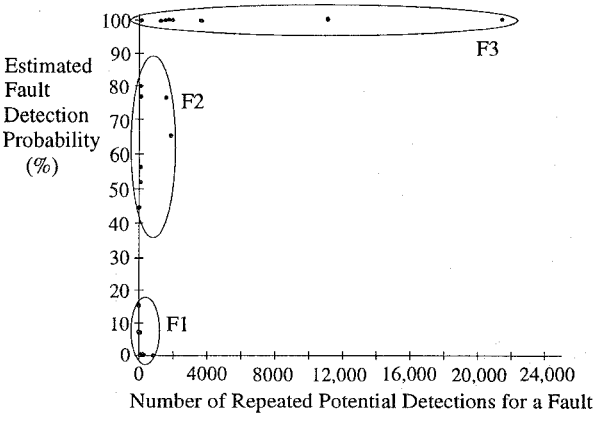

Figure 4: Estimated fault detection probability vs. number of potential detections: s5378

F1 in Figure 3) that are potentially detected over 11,000 times are never detected for any of the 100 random initial states. One fault (marked F2 in Figure 3) is potentially detected 7476 times but is only detected for 42% of the initial states. Several of the faults which are detected for all 100 of the initial states (marked F3 in Figure 3) are potentially detected fewer times than faults in the F1 and F2 groups. For s5378, many of the faults potentially detected a relatively small number of times are detected for a small percentage of the random initial states (marked F1 in Figure 4). However, several faults potentially detected a small number of times are detected for most or all of the initial states (marked F2 and F3 in Figure 4). Similar results were obtained for several other circuits.

Thus, the number of times a fault is potentially detected during fault simulation is not a good indicator of the fault detection probability. A high potential detection count does not guarantee a high probability of detection, and faults that are potentially detected only a small number of times may in fact be detected by the tester 100% of the time. To replace the number of potential detections as a measure of fault detection, which can be computed very quickly as a by-product of conventional fault simulation, we need another measure that will be more accurate but similarly easy to compute. We propose in this work the use of the estimated fault detection probability for this purpose.

To establish the accuracy of the estimated fault detection probability in identifying faults that will be detected during the test application process, we compare the results obtained using the estimated fault detection probability to the results obtained by accurate simulation. Accurate logic simulation using high-level descriptions was proposed in [12]. Restricted symbolic simulation has been proposed in [13, 14], in which individual unknown symbolic values are assigned to circuit nodes. The fault simulation procedures of [15, 16, 17] can also be used to obtain accurate fault detection information. We use in this work the procedure from [17] and additional procedures described in Section IV.

III Fault Detection Probability Estimation Tool

In accordance with the discussion above, we restrict our attention to the testing technique referred to as the restricted multiple observation time strategy; that is, we use a unique good machine output sequence, computed by starting the good machine from the all-unknown initial state. For the faulty circuit, we compute 100 output sequences, starting from 100 randomly selected fully-specified initial states. Thus, for a given fault simulation run, the good machine initial state is

kept an unknown and the faulty machine initial state is kept a fully-specified random vector. This guarantees a unique good machine output sequence for our hundred different fault simulation runs.

As an aside, a good machine may have unique, deterministic output values that cannot be determined using 3-value simulation, but can be proven by symbolic or exhaustive state simulation. This could increase the number of potential detections and may also increase the probability of detection. Moreover, additional faults may be detected by using the multiple observation time approach [18]. In our study, we did not exploit these features.

The PROOFS sequential circuit fault simulator [2] was modified to allow for the specification of the initial state for the faulty circuits. The same initial state is used by all faulty circuits so that up to 32 faults can be simulated in parallel using the 32-bit parallelism of the computer word. The PROOFS fault simulator combines features of bit-parallel, differential, and concurrent fault simulation algorithms. For each test vector, the good circuit is first simulated, and then only differences between the good and faulty circuits are simulated. Several faulty circuits are simulated together, with one bit of the computer word used for each faulty circuit, and faults are grouped dynamically with each test vector simulated, in order to fully utilize all bits in the computer word. To limit the memory usage, faulty circuit values are stored at the flip-flops only. For each faulty circuit, a linked list of state lines that have different values from the good circuit is maintained. Before a group of faults is simulated, the circuit structure is modified to emulate the effects of the faults, and the state line values are updated. In the modified PROOFS, the specified initial state is also updated for the first test vector. Faults are declared to be detected when a 1/0 or 0/1 value pair for the good and faulty circuits reaches a primary output. Faults are dropped from the fault list once they are detected, and faults that are identified as inactive in a given time frame are not simulated.

An executive program was developed to generate a specified number of random initial states, invoke the fault simulator with each initial state, and compute statistics on the fault simulation runs. For each fault simulated, the number of runs for which the fault was detected is provided. The estimated fault detection probability is then computed as the fraction of runs for which the fault is detected.

In some fault simulators, it may not be feasible to set the initial state of the faulty machine independently of the initial state of the good machine, and the two states may have to be specified to the same value. This causes a problem, since it results in a fully-specified response of the good machine, when in fact the response may be different for different initial states, and should be represented by unknown values. As a result, faults that are not detected may appear to be detected. To solve this problem, we observe that if the good circuit is synchronized by the test sequence being simulated, then after several vectors are applied, the good machine output values become fully specified, independent of the initial state. We will call the period from the beginning of the test sequence to the time when values become known the *transient period*. The length of the transient period can be computed by first performing simulation starting from the all-unknown initial state. During fault simulation starting from fully-specified initial states, faults that get detected only during the transient period must be discounted to avoid the inaccuracy mentioned earlier. In our experiments, we did not face this problem, since we had the ability to set the good machine state to an unknown value while fully specifying the faulty machine state.

IV Validating the Probabilistic Detection Methodology

For the estimated fault detection probability computed using a random sample to be useful, it must be close to the value we would obtain by including all possible initial states in the sample. For the sample size of 100 used in this work, from standard statistical methods it can be surmised that at a 99% confidence level, the error margin in the estimated fault detection probability is the highest (approximately ±12%) when the detection probability is 50%. When the detection probability is 90%, the error margin in the estimated fault detection probability is ±7.7%, and for 99% detection probability, the error is -2.5% to +1%. The error can be halved by increasing the sample size to 400.

The question remains whether faults identified as detected for 100% of the random initial states can be guaranteed to be detected by the tester. Accurate symbolic simulation is currently infeasible for large circuits. Therefore, we have developed several tools to verify the detection of such faults. The first tool does parallel restricted symbolic fault simulation using a modification of PROOFS [2]. The second tool uses a variation of the fault detection probability estimation tool to do fault simulation starting from an exhaustive list of partially-specified states. The third tool is the one from [17]. These tools are not intended to replace the fault detection probability estimation tool but rather to validate the proposed methodology of probabilistic detection.

A Parallel restricted symbolic fault simulation

Restricted symbolic simulation was proposed in [13, 14]. With this approach, specific named unknowns are assigned to lines that initially have unknown values. This approach was referred to as *individual initial value propagation* in [13] and was applied to faults in the clock control logic. Symbolic simulation is useful in situations where a circuit cannot be initialized using 3-value logic. Consider the circuit shown in Figure 5. A specific named unknown, Y, is assigned to the flip-flop

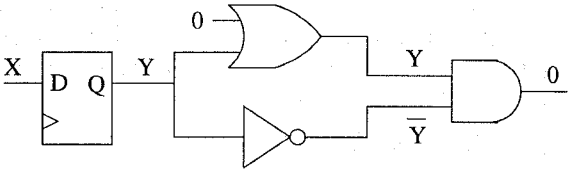

Figure 5: Symbolic initialization example

output. The resulting values propagated to the AND-gate inputs are Y and $\overline{Y}$. The AND-gate can therefore be evaluated to logic 0, whereas the value would be unknown if 3-value logic was used. This situation can arise during fault evaluation, and the symbolic approach can aid in identifying detected faults. Restricted symbolic simulation is still not as accurate as full symbolic simulation. For example, if the inputs of an AND-gate are two different named unknowns, the output is unknown, X. Thus, useful information may be lost, but the restricted symbolic approach is much more efficient.

Symbolic simulation is useful in evaluating fault effects, even when it does not lead to circuit initialization. Consider the circuit shown in Figure 1. The good circuit is easily initializable by setting the MUX control line to 1 and setting the second input of the MUX to 0 or 1. However, the faulty circuit cannot be initialized under 3-value simulation. Assigning a named unknown, Y, to the faulty circuit flip-flop output in the time frame in which the good circuit flip-flop is initialized enables the generation of 0/Y and 1/Y good and faulty circuit value pairs at the flip-flop output. This situation is illustrated in Figure 6, in which values for 3 time frames are listed for each node. The Y value represents the logic value of the flip-flop at a specific point in time. Y is either a 1 or a 0. If both 1/Y (or $0/\overline{Y}$) and 0/Y (or $1/\overline{Y}$) value pairs are propagated to the primary outputs, the fault is known to be detected. The fault effects may appear in different time frames and on different primary outputs, as long as the Y refers to the value of the flip-flop at the same point in time.

Rather than using several named unknowns to represent the values of multiple state lines as is done in

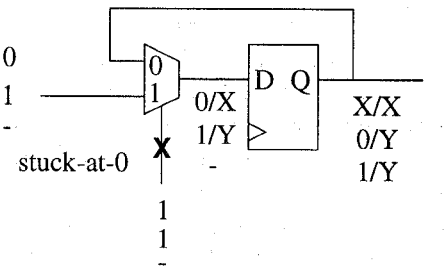

Figure 6: Generation of 1/Y and 0/Y value pairs

[13, 14], we use a single named unknown, Y. A Y is placed at the first flip-flop evaluated for which the good circuit value is known but the faulty circuit value remains unknown. Faults on the reset lines of single flip-flops can use this approach effectively. However, for some faults on the reset lines of multiple flip-flops, the fault effects can only be propagated effectively through certain flip-flops. In our restricted symbolic fault simulator, we allow the user to specify a flip-flop for placement of the named unknown.

The PROOFS sequential circuit fault simulator [2] was used to implement the symbolic evaluation tool. PROOFS evaluates up to 32 faults in parallel, and thus the resulting tool is a parallel restricted symbolic fault simulator. PROOFS was modified to include the values Y and $\overline{Y}$ in addition to 0, 1, and X. Faulty circuit evaluation routines were modified accordingly, and flags were added for the 1/Y ($0/\overline{Y}$) and 0/Y ($1/\overline{Y}$) value pairs for each fault. A Y value is placed at a flip-flop if the good circuit value is known, the faulty circuit value is unknown, and no Y value exists at another flip-flop for that fault. If the Y value does not propagate to the flip-flops and the fault is not identified as detected and the faulty circuit flip-flop value remains unknown, a new Y value will have to be used in the next time frame. In this case, the flags must first be reset to 0. The parallel restricted symbolic fault simulator is accurate in the faults that it identifies as detected. However, it may miss some faults due to the loss of information as compared to full symbolic fault simulation.

B Implicit exhaustive fault simulation

For circuits having a small number of flip-flops, such as s820 and s832, fault simulation can be performed starting from all possible fully-specified states to verify the results of the fault detection probability estimation tool. For other circuits, the number of states is too large, but for many of these circuits, all possible initial states can be implicitly tried by using partially-specified states. Some flip-flops are initialized to 1 or 0 values, while the remaining flip-flops are assigned X values. For the flip-flops with known values, all possible

combinations of values are tried. We refer to this approach as *implicit exhaustive fault simulation.* This approach is based on the one proposed in [15].

A tool was developed to do implicit exhaustive fault simulation using the modified PROOFS fault simulator developed for the fault detection probability estimation tool. The user must specify which flip-flops have to be initialized to known 1 or 0 logic values in order for the fault effects to be propagated to the primary outputs. For each partially-specified initial state, the modified PROOFS is invoked to determine if a given fault is detected. Statistics are computed on the number of runs for which each fault is verified to be detected. The flip-flops requiring known values must be selected carefully. Detection of a fault for fewer than 100% of the partially-specified initial states does not mean that a fault will not be detected for 100% of the fully-specified states. It is possible that the X value propagated from a flip-flop masks the propagation of fault effects, and thus the fault is missed due to the loss of information inherent in 3-value simulation. We use a trial-and-error procedure to identify the flip-flops for initialization. This approach may be ineffective for circuits having large state spaces.

V Results

A fault detection probability estimation tool was implemented using the existing PROOFS [2] fault simulator. The tool was used to evaluate the potentially detected faults for several of the ISCAS89 sequential benchmark circuits. Test sets generated by HITEC [11] were used, first to identify the potentially detected faults, given the collapsed fault lists for the circuits, and then to perform fault simulation for specific initial faulty circuit states. Evaluations were carried out on an HP 9000 J200 with 256 MB RAM.

Results of fault simulation starting from 100 random fully-specified states are shown in Table 1. For each of the 100 fault simulation runs, the faulty circuit states were initialized with the given random vector, while the good circuit state was kept an unknown. For each circuit, the number of test vectors, the number of flip-flops, the total number of faults, and the number of potentially detected faults are given, followed by the number of potentially detected faults having estimated detection probabilities of 0%, 100%, and ranging between 1% and 20%, 21% and 40%, 41% and 60%, 61% and 80%, and 81% and 99%. The execution time required to complete the 100 fault simulation runs is shown in the last column; times ranged from eighteen seconds to 24 minutes. The ratio of the execution time required for the 100 fault simulation runs to the execution time required for one fault simulation run using the full fault list ranged between 13 for the largest circuit and 61 for one of the smaller circuits having a short test set (s641). Relatively larger run times are required for fault simulation of small circuits using short test sets because the work of initializing the data structures is done repeatedly, and this work is a large portion of the fault simulation processing. Accuracy can be traded off for execution time by varying the random sample size. In some circuits, such as s298, all the potentially detected faults have an estimated fault detection probability of 100%. In other circuits, there are potentially detected faults with 0 or very low estimated fault detection probabilities. For example, s5378 has 11 faults with an estimated fault detection probability of 0, and 34 faults with estimated fault detection probabilities in the range 1–20%.

Several faults having high detection probabilities in the range 70%–100% are identified using the new approach. In practice, these faults have a very high probability of being detected by a tester. In contrast, if any faults having low detection probabilities less than 20% occur on the physical device, they will likely escape detection. Faults with very low detection probabilities are of more concern than redundant faults. If a redundant fault occurs on a device, it has no effect on the correctness of the function. However, if a fault having low detection probability occurs, it will have an effect on the function at unexpected times. The logic should therefore be resynthesized or design-for-testability (DFT) hardware should be added to remove these faults. The DFT approach described in [19] applies to faults affecting lines involved in setting or resetting flip-flops. However, in our experiments, we observed that most such faults had very high detection probabilities and therefore are not candidates for the DFT methodology of [19]. For faults having low detection probabilities, additional controllability and observability may be inserted by methods presented in [20]. In addition, synthesis-for-testability techniques can also be used [21, 22].

Procedures to verify the detection status of faults having estimated detection probabilities of 100% were implemented, as described in Section IV. Results are shown in Table 2. The parallel restricted symbolic fault simulator was first run, and a large number of the faults were identified as detected (**Y Sim**). For one fault in each of s344 and s349, the symbolic value had to be assigned to a specific flip-flop in order for the fault to be detected. For faults on reset lines of single flip-flops, this problem is not encountered. For any faults not identified as guaranteed detected by the first procedure, implicit exhaustive fault simulation was attempted on a subset of the flip-flops. To facilitate this procedure,

Table 1: Estimated Fault Detection Probabilty for 100 Random Initializations

Circuit	Vectors	Flip-Flops	Total Faults	Pot Det Faults	Faults with Est. Detection Probability (%)							CPU Time
					0	1–20	21–40	41–60	61–80	81–99	100	
s298	322	14	308	8	0	0	0	0	0	0	8	22.3s
s344	127	15	342	6	0	0	0	0	0	0	6	18.0s
s349	134	15	350	6	0	0	0	0	0	0	6	18.0s
s382	2074	21	399	15	2	0	0	1	0	0	12	1.38m
s400	2214	21	426	15	2	0	0	1	0	0	12	1.45m
s444	2240	21	474	15	2	0	1	0	1	0	11	1.05m
s526	2258	21	555	19	3	1	0	1	0	0	14	1.73m
s641	209	19	467	9	0	0	0	3	2	0	4	32.4s
s713	173	19	581	9	1	0	0	2	2	0	4	32.3s
s820	1115	5	850	1	0	0	0	0	0	0	1	1.03m
s832	1137	5	870	1	0	0	0	0	0	0	1	1.05m
s953	23	29	1079	82	19	37	1	0	9	10	6	33.9s
s1423	89	74	1515	38	9	0	1	18	0	1	9	38.7s
s1488	1170	6	1486	2	0	0	0	0	0	0	2	1.94m
s1494	1245	6	1506	2	0	0	0	0	0	0	2	2.04m
s5378	912	179	4603	69	11	34	0	4	4	0	16	6.90m
s35932	496	1728	39094	10	0	0	0	0	0	0	10	24.2m

a list of flip-flops that must be initialized to known 1 or 0 logic values in order for the fault to be detected was required. A trial-and-error procedure was used to obtain this list of flip-flops, and the method was not used for some faults in the largest circuits, either due to the large state space or difficulty in identifying the required flip-flops. Most of the remaining faults were proven to be detected using this approach (**Impl Exh**). More accurate simulation was used to verify the detection status of the remaining faults ([17]).

All except one of the faults having an estimated detection probability of 100% for an initial state sample size of 100 were verified to be detected using various approaches. One fault in s1423 was not proven to be detected; however, it has a very high probability of detection during the process of test application to the physical circuit. The fault detection probability estimation tool is straightforward to implement, results are accurate, and execution times are moderate.

VI Conclusion

Certain faults, such as faults on the reset lines of flip-flops, cause a circuit to remain in an unknown state, which prevents a fault simulator from identifying the faults as detected. Instead, the faults are found to be potentially detected. Common industrial practice is to declare the faults to be detected after several repeated potential detections. However, results show that this approach is misleading; faults that are potentially detected several times starting from an unknown state may have a very low probability of detection, and faults that are potentially detected only a small number of times starting from an unknown state may be detected with a very high probability. A better approach to estimating the fault detection probability is to perform several fault simulations starting from different random fully-specified states. Faults having estimated detection probabilities of 100% using this approach were verified to be detected by more computation-intensive approaches. Furthermore, execution times are moderate, and accuracy can be traded off for execution time.

Acknowledgment

We would like to thank Dr. Bill Underwood of Sunrise Test for useful discussions on restricted symbolic fault simulation.

References

[1] W.-T. Cheng and M.-L. Yu, "Differential fault simulation - A fast method using minimal memory," *Proc. Design Automation Conf.*, pp. 424–428, 1989.

[2] T. M. Niermann, W. -T. Cheng, and J. H. Patel, "PROOFS: A fast, memory-efficient sequential circuit fault simulator," *IEEE Trans. Computer-Aided Design*, pp. 198–207, February 1992.

[3] N. Gouders and R. Kaibel, "PARIS: A parallel pattern fault simulator for synchronous sequential circuits," *Proc. Int. Conf. Computer-Aided Design*, pp. 542–545, 1991.

[4] H. K. Lee and D. S. Ha, "HOPE: An efficient parallel fault simulator for synchronous sequential circuits," *Proc. Design Automation Conf.*, pp. 336–340, June 1992.

Table 2: Proven Detection of Potentially Detected Faults

Circuit	Faults				
	Pot Det	Estimated 100% Det	Proven Detected		
			Y Sim	Impl Exh	[17]
s298	8	8	7	1	-
s344	6	6	5	1	-
s349	6	6	5	1	-
s382	15	12	11	1	-
s400	15	12	11	1	-
s444	15	11	10	1	-
s526	19	14	12	2	-
s641	9	4	4	-	-
s713	9	4	4	-	-
s820	1	1	0	1	-
s832	1	1	0	1	-
s953	82	6	0	6	-
s1423	38	9	5	2	1
s1488	2	2	1	1	-
s1494	2	2	1	1	-
s5378	69	16	10	2	4
s35932	10	10	9	0	1

[5] K. Cheng and V. Agrawal, "State assignment for initializable synthesis," *Proc. Int. Conf. Computer-Aided Design*, pp. 212-215, 1989.

[6] K. Cheng and V. Agrawal, "Initializability consideration in sequential machine synthesis," *IEEE Trans. Computers*, vol. 41, pp. 374-379, March 1992.

[7] J. A. Wehbeh and D. G. Saab, "On the initialization of sequential circuits," *Proc. Int. Test Conf.*, pp. 233–239, 1994.

[8] K.-T. Cheng, "On removing redundancy in sequential circuits," *Proc. Design Automation Conf.*, pp. 164–169, 1991.

[9] T. J. Powell, "Consistently dominant fault model for tristate nets," *Proc. VLSI Test Symp.*, pp. 400–404, 1996.

[10] I. Pomeranz and S. M. Reddy, "Classification of faults in synchronous sequential circuits," *IEEE Trans. Computers*, pp. 1066–1077, Sept. 1993.

[11] T. M. Niermann and J. H. Patel, "HITEC: A test generation package for sequential circuits," *Proc. European Conf. Design Automation (EDAC)*, pp. 214–218, February 1991.

[12] S. J. Chandra and J. H. Patel "Accurate logic simulation in the presences of unknowns," *Proc. Int. Conf. Computer-Aided Design*, pp. 34–37, 1989.

[13] T. Ogihara, S. Saruyama, and S. Murai, "Test generation for sequential circuits using individual initial value propagation," *Proc. Int. Conf. Computer-Aided Design*, pp. 424–427, 1988.

[14] J. L. Carter, B. K. Rosen, G. L. Smith, and V. Pitchumani, "Restricted symbolic evaluation is fast and useful," *Proc. Int. Conf. Computer-Aided Design*, pp. 38–41, 1989.

[15] I. Pomeranz and S. M. Reddy, "Fault simulation for synchronous sequential circuits under the multiple observation time testing approach," *Proc. European Test Conf.*, pp. 292–300, 1993.

[16] R. Krieger, B. Becker, and M. Keim, "A hybrid fault simulator for synchronous sequential circuits," *Proc. Int. Test Conf.*, pp. 614–623, 1994.

[17] I. Pomeranz and S. M. Reddy, "Low-complexity fault simulation under the multiple observation time testing approach," *Proc. Int. Test Conf.*, pp. 272–281, 1995.

[18] I. Pomeranz and S. M. Reddy, "The multiple observation time test strategy," *IEEE Trans. Computers*, pp. 627–637, May 1992.

[19] M. Abramovici and P. S. Parikh, "WARNING: 100% fault coverage may be misleading!!" *Proc. Int. Test Conf.*, pp. 662–668, 1992.

[20] E. M. Rudnick, V. Chickermane, P. Banerjee, and J. H. Patel, "Sequential circuit testability enhancement using a nonscan approach," *IEEE Trans. VLSI Systems*, vol. 3, no. 2, pp. 333–338, June 1995.

[21] I. Pomeranz and S. M. Reddy, "Design and synthesis for testability of synchronous sequential circuits based on strong-connectivity," *Proc. Fault-Tolerant Computing Symp.*, pp. 492–501, 1993.

[22] I. Pomeranz and S. M. Reddy, "On removing redundancies from synchronous sequential circuits with synchronizing sequences," *IEEE Trans. Computers*, pp. 20–32, Jan. 1996.