Fault Sampling and Statistical Fault Simulation

Adam A. Ray

***Abstract*—Techniques and theories regarding statistical fault simulation are presented. The literature regarding statistical fault simulation is explored from its beginning ideas to present-day applications. Results from different algorithms and applications found in the literature are given. As an example of fault sampling, the affects of fault sampling for fault coverage estimation is shown on the s35932 benchmark circuit. For this example, as the sample size increases, the error decreases until the estimated coverage approaches the true fault coverage.**

I. Introduction

As the complexity of today's VLSI chips increases, the complexity of the testing of these chips increases exponentially. The number of transistors per chip has reached in the upper millions making it impossible to exhaustively test all faults in a circuit. In an effort to reduce the complexity of the testing of these chips, fault sampling was introduced into VLSI testing.

Fault sampling was first used to estimate the coverage of a given set of test vectors in a fraction of the time it would take using a complete fault list. In a typical application, a subset of faults is randomly selected from a set of all possible faults. Usually the subset is much smaller than the total fault list. This subset of faults is simulated to produce an estimated fault coverage known as the sample coverage [3]. As is pointed out and proven by many authors, the sample size of the subset should be selected based on the desired confidence or accuracy of the sample coverage. It need not be based on the size of the original population [7], or in this case, the total number of faults.

From here, fault sampling has seen almost 30 years of evolution into a world of ideas and applications. Statistical fault simulation is a term that is used by this author to summarize this "world of applications"; of which simple fault sampling is just the beginning. This paper attempts to give an introduction to the ideas and background of statistical fault simulation.

The rest of this paper is organized as follows: Section II traces the major developments of statistical fault simulation from start to present. Section III gives an example of fault sampling using the s35932 benchmark circuit. Section IV provides results from different algorithms found in the literature. Section V concludes this paper.

II. Background

The earliest work proposing fault sampling for test sequence evaluation in digital circuitry is [4]. In this paper, faults are selected at random and fed into a fault simulator. The number of non-detectable faults is plotted against the number of faults simulated (defined as the sample size in later works). As faults are simulated, the graph is traced into an "accept" or "reject" region.

In [1], Agrawal divided the ideas of fault sampling into two defined approaches: fixed sample size method and sequential sampling. Sequential sampling involves simulation of several small sets of sampled faults, each resulting coverage estimate brining you closer to the actual value. This approach also provides a way to know the coverage range (p0-p1) that bounds the estimated fault coverage, instead of a simple "accept/reject" result as in Case's approach. Equations are given for calculating the fault coverage from fault sampling with a 99% confidence level in another work [2].

The fixed sample size method, as the name implies, involves simulation of a predetermined and set number of faults. It is shown that as the sample size approaches 1000 or more, the fault coverage asymptotically approaches the true coverage for any circuit [1]. The reason for this is that the number of sampled faults depends only on the desired confidence range and not the size of the circuit. This paper also points out that the simulation of several small samples of faults simultaneously, as in sequential sampling, is ideal for parallel simulators.

STAFAN (STAtistical Fault ANalysis) is presented in [13] as an alternative method to fault sampling. Typical fault sampling is problematic because it assumes the availability of a complete fault simulation program. This is not always the case because chip technology usually leads simulation technology. Typical fault sampling also gives no information about the faults that are not part of the sample, even though an "undetectable list" is often desired. STAFAN uses randomly sampled test vectors (not sampled faults, but sampled test vectors) to calculate controlabilities and observabilities, which are redefined in terms of statistical activity on a line. Since the probability of detecting a fault is the joint probability of controlling a line to a number and observing the line on an output, the product of the controllability and observability of a line yields the probability of detection of a fault on that line. Faults with high probability are the ones likely to be detected. From here, a simple calculation yields the total fault coverage.

This approach is one of the first methods with sequential as well as combinational abilities. Also, it only involves true-value (fault-free) simulation; therefore, its complexity grows only linearly with the number nodes, as opposed to the exponential growth associated with conventional statistical simulation methods.

McNamer et al applied much of the detailed theory of

statistics to sampling and fault simulation [15]. This is the first paper to provide equations for fault coverage with a 100% confidence level. Simulation-based fault sampling is divided into two categories: simple and stratified. The previously defined methods of fixed sample size and sequential sampling fall into the simple sampling category. The stratified category comes from a geology word, "strata", meaning deposited layers. In stratified sampling, the sample space is divided into smaller subspaces such that the faults in each stratum share some common trait, much like the layers of rock deposited in geology. From here samples are taken from each stratum, and a weighted sum leads to the overall fault coverage.

In [16], a relationship between fault coverage and probabilistic testability is established. This relationship is then used to propose a method of test generation using randomly sampled faults. This is the first work regarding statistical test generation. The generated test vectors from these randomly sampled faults are used to estimate the overall fault coverage.

Ideas of fault sampling move to test vector sampling in [11]. This method uses true-value simulation of a circuit with randomly sampled test vectors to estimate the controlabilities and observabilities of all circuit lines, and therefore, all faults. These estimations are used to compute the controlabilities and observabilities based on the entire vector set, which are used to compute the fault coverage much like in [13].

This method may be used on combinational circuits and it is shown to work on stuck-at and delay faults. The complexity of the algorithm is a function of the number of circuit lines only, whereas [1] is a function of the number of circuit lines and the number of test vectors. A fault simulator without sampling is a function of the number of circuit lines, test vectors, and faults. Again, the number of sampled vectors depends on the desired coverage accuracy, not on the total number of lines, vectors, or faults. This method could work especially well for built in self test (BIST) because the typical number of test vectors can be in the millions.

Constantinescu takes the sampling approach into what he calls a "multi-dimensional state of events" [5][6]. He proposes that there are three major things that affect a fault-induced error and its propagation through the circuit (and, therefore, the fault coverage). These are 1) fault location and type, 2) timing of the fault, and 3) workload executed at the time of the fault occurrence. Typical fault sampling methods only take the first into account. He uses N-stage, stratified sampling. "N-stage" involves dividing sampled subpopulations (or strata) into more sampled subpopulations N levels deep. His 3-stage sampling method takes into account the three things listed above, representing the workload by the number of active system inputs. His method is evaluated using four different hypothetical circuits [6].

In [9] a new method for estimating the Defect Level (DL) using stratified sampling on defect-oriented (DO) extracted faults is presented. This paper also proposes a new method for DO fault extraction. DL is defined in [3] as the fraction of faulty chips among the chips that pass a test.

Several different statistical methods for defining confidence regions are given in [8]. Frequency and Bayesian methods are compared for both simple and stratified sampling. Interestingly, stratification is shown to degrade the coverage confidence limits in this paper.

Goncalves et al [10] deal with fault sampling under tight safety priorities and regulations, cases in which a minimum fault coverage of 100% is desirable. The paper introduces the term "Property Coverage" (PC) which is an extension of fault coverage to include information such as timing, subsets of outputs, and multiple fault analysis. Very promising results are given.

Thaker, Agrawal, and Zaghloul propose a procedure to estimate gate-level coverage of test patterns using stratified fault sampling and register-transfer level (RTL) fault simulation [18]. The RTL coverage is experimentally shown to track the gate coverage in this paper. Therefore, the complete RTL fault list can be treated as a sample of the gate-level fault list such that each RTL module fault list is a stratified sample. The overall fault coverage is calculated as a weighted sum of individual RTL module coverages using the stratified sampling technique. This approach is shown to out-perform gate-level simulation approaches in CPU time and memory. The problem is that most simulators are optimized for gate-level fault models, and not RTL fault models. It shows much future promise.

III. Analysis of Fault Sampling

To explore the ideas behind fault sampling, a simple example is constructed in which the s35932 benchmark circuit is simulated with fault sampling for several different sample sizes. The sequential circuit has 35 inputs, 320 outputs, and 39,094 collapsed faults. Simulations and test sequence generations were performed using the Hitec/Proofs software [12] designed by the University of Illinois. Random selection of faults was made possible using Matlab and Excel. An example of a simulator which has built-in random fault injection capabilities is described in [17].

The notation used is adopted from [3]:

C=true fault coverage (typically unknown)
c=sample coverage (random variable)
x=value of c determined from simulation
Ns=sample size

A test sequence of 222 vectors was generated for the s35932 circuit, yielding a true fault coverage, C, of 0.8919. The circuit was simulated using eight different sample sizes. The results are given in Table 1. The simulations were run on a Pentium-IV, 2.4 GHz processor running Windows XP. The 3σ error bounds are calculated using the equation given in [3]:

$$3\sigma_error = \pm\frac{4.5}{Ns}\sqrt{1+0.44*Ns*x*(1-x)} \qquad (1)$$

Ns	x	3σ bounds	Time (sec)
39,094	0.8919	±0.00	95
4000	0.9050	±0.0142	21
2000	0.8860	±0.0213	20
1000	0.8990	±0.0288	17
500	0.8900	±0.0427	17
250	0.8920	±0.0613	14
100	0.8800	±0.1069	14
50	0.9400	±0.1347	15
25	0.1000	±0.1800	14

Table 1: Circuit s35932 was simulated with random fault sampling for several sample sizes, Ns. The estimated fault coverage, x, is given along with the 3σ error bounds and computation time. Notice the large computation time difference of the simulation of all 39,094 faults compared to the simulation of any set of sampled faults.

Using the values in Table 1, a percent error is calculated and plotted in Figure 1 against the sample size, Ns. The error grows as Ns approaches zero. In this example, for Ns>100, the error oscillates about zero. Typically, this happens for Ns>1000 or more [1]. The percent error is calculated by

$$\%\,Error = \frac{(x - C)}{C} * 100\,\% \qquad (2)$$

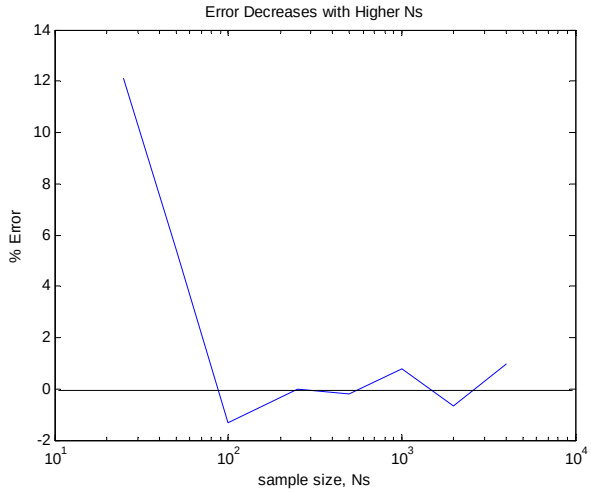

Figure 1: Percent Error is plotted as a function of Ns. As Ns approaches zero, the error increases. For Ns>100, the error oscillates about zero.

IV. Results from Different Algorithms

To explore statistical fault simulation deeper, we turn to results from the literature. The STAFAN algorithm [13] results are shown in Table 2 for five different circuits. The average deviations of STAFAN from the true coverage are between 0.29 and 3.65%. In another work [19], the algorithm's CPU time is shown to increase only linearly with the number of gates (for combinational circuits). This is compared to a concurrent fault simulator [14] which has a nonlinear complexity. The results are shown in Table 3. With the combinational circuits, STAFAN had a maximum error of only 3% in the estimation of the fault coverage.

Circuit	# of vec.	# of faults	N	Final Coverage		
				C	x	|err|
4-bit ALU	52	263	10	0.9657	0.9625	0.0047
64-bit ALU	155	4376	20	0.7530	0.7509	0.0029
4-bit Mult.	1111	741	100	0.9312	0.8680	0.0337
Circuit A	3842	5060	200	0.8644	0.8361	0.0124
Circuit B	3636	5856	200	0.7126	0.7257	0.0365

Table 2: Results from the STAFAN algorithm. N is the number of test vectors applied. C is the true fault coverage from fault simulation. x is the fault coverage estimated by STAFAN. The last term, |err|, is the average absolute deviation between x and C.

Circuit Name	No. of Gates	No. of Vectors	Concurrent Cov.(%)	Concurrent CPU s	Stafan Cov. (%)	Stafan CPU s
alu4	106	27	97.0	10	97.7	15
c432	244	68	90.0	45	88.9	22
c880	670	72	100	84	99.8	45
c1355	706	154	99.3	135	95.8	41
c1499	463	75	93.9	43	93.4	36
c1908	1162	211	99.7	326	98.2	110
c3540	2735	218	96.1	1097	93.5	309
c6288	2484	33	99.2	2062	99.4	428
s344	339	112	81.5	131	72.9	49
s382	397	1726	79.1	2564	63.1	323
s526	432	2750	70.3	7032	51.2	523
s641	628	124	73.4	198	71.7	65
s1196	746	338	86.9	544	85.9	149
s1488	732	401	89.5	604	90.3	168
s1498	726	1144	58.4	920	69.77	316
s5378	4788	2335	62.58	15264	59.9	3589
4ff	33	16	61.7	-	42.6	-

Table 3: STAFAN algorithm compared to a concurrent simulator in terms of fault coverage and CPU time for a variety of combinational and sequential benchmark circuits.

Results from [16] are shown in Table 4. Fault sample sizes of 500, ~1000, and ~1500 are used for test vector generation which is in turn used to estimate the fault coverage. For each of the three circuits in Table 4, the estimated coverage approaches the measured coverage as the sample size increases. As is expected, the number of generated vectors (and therefore the computational time) increases with the increased sample size.

In [10], an ASIC gas burner control system (GBCS) was tested with the new fault sampling method presented in the paper. In safety regulated applications such as this, 100% property coverage (PC) is desired. PC is a new term which extends the ideas of fault coverage into a broader range of applications. The total numbers of faults in the GBCS along with the chosen sample sizes are shown in Table 5. Table 6 shows the results from simulation of these sampled faults. All of the simulated faults were either fault tolerant or fail-safe, yielding the desired PC of 100% with a confidence level of >90%. See [10] for confidence level calculations.

Sample Size	Circuit Name →	C2670	C6288	C7552
	Total Faults →	2747	7744	7550
500	Vectors →	65	34	77
	Adjusted sample size →	488	500	496
	Sample Cov. (%) →	100.0	100.0	100.0
	Estimated Cov. (%) →	94.0	97.0	93.3
	Measured Cov. (%) →	–	98.4	–
	Test gen. CPU Sec. →	1300	20	698
	Fault Sim. CPU Sec. →	14	20	45
978	Vectors →	96		
	Sample Cov. (%) →	100.0		
	Estimated Cov. (%) →	96.4		
	Measured Cov.(%) →	97.2		
	Test gen. CPU Sec. →	2900		
	Fault Sim. CPU Sec. →	20		
1485	Vectors →			142
	Sample Cov. (%) →			100.0
	Estimated Cov. (%) →			95.9
	Measured Cov. (%) →			95.6
	Test gen. CPU Sec. →			2855
	Fault Sim. CPU Sec. →			78

Table 4: Test generation by fault sampling. Three circuits are tested for three different fault sample sizes.

	Single Faults		Double Faults	
	SA	**BRI**	**SA**	**BRI**
Total faults	2,818	4,575,312	3,967,065	10^{13}
Ns	--	4,740	18,039	2,192

Table 5: Total number of faults and sample sizes used in the GBCS example in [10]. SA=stuck-at fault. BRI=bridging fault. Ns=sample size.

Proprieties	Double SAs		Single BRIs		Double BRIs	
Fault tolerant	4,859	(26.94%)	1,607	(33.90%)	392	(17.88%)
Fail-safe	13,180	(73.06%)	3,133	(66.10%)	1,800	(82.12%)

Table 6: Simulation results of the GBCS example using sample sizes in Table 5. 100% PC is achieved.

V. Conclusion

Theories and Applications in the field termed as statistical fault simulation have been explored and reported. The background of fault sampling has been presented along with its present-day applications. An example of the affects of fault sampling was shown on the s35932 benchmark circuit. The results were as expected and agreed with other results in the literature, except in one respect. Theoretical calculations yield a sample size of 1000 or greater for confident fault coverage estimation. This is also generally true for other examples from the literature. However, in this particular example, the estimated coverage approached the true coverage after only 100 samples were used. Perhaps more simulations must be performed since it is a probabilistic measure.

Results from several algorithms in the literature have been given. Several of these algorithms have promise of further development in the future. Test vector sampling ideas of [11] have high promise of future use due to a rise in the application of BIST, which can require millions of test vectors.

References

[1] Agrawal, V. D., "Sampling techniques for determining fault coverage in LSI circuits," Journal of Dig. Sys., vol. 5, no. 3, pp. 189-202, 1981.

[2] Agrawal, V. D., Hatsuyoshi, K., "Fault sampling revisited," IEEE Design & Test of Computers, pp. 32-35, 1990.

[3] Bushnell, M. L., Agrawal, V. D., *Essentials of Electronic Testing for Digital,, Memory and Mixed-Signal VLSI Circuits*. New York: Springer, 2000.

[4] Case, G. R., "A statistical method for test sequence evaluation," Proc. 12th Conf. Design Automat., pp. 257-260, 1975.

[5] Constantinescu, C., "Inferring coverage probabilities by optimum 3-stage sampling," Proc. Reliability and Maintain. Symp., pp. 61-65, Jan 1996.

[6] Constantinescu, C., "Using multi-stage & stratified sampling for inferring fault-coverage probabilities," IEEE Trans. Reliability, vol. 44, no. 4, pp. 632-639, December 1995.

[7] Cooper, G. R., McGillem, C. D., *Probabilistic Methods of Signal and System Analysis*. New York: Oxford University Press, 1999.

[8] Cukier, M., Powell, D., Arlat, J., "Coverage estimation methods for stratified fault-injection," IEEE Trans. Computers, vol. 48, no. 7, pp. 707-723, July 1999.

[9] Goncalves, F. M., Santos, M. B., Teixeira, I. C., Teixeira, J. P., "Defect-oriented test quality assessment using fault sampling and simulation," IEEE Int. Test Conf., pp. 35-42, October 1998.

[10] Goncalves, F. M., Santos, M. B., Teixeira, I. C., Teixeira, J. P., "Self-checking and fault tolerance quality assessment using fault sampling," Proc. 17th Inter. Symp. Defect & Fault Toler. in VLSI Sys., pp. 216-224, Nov. 2002.

[11] Heragu, K., Agrawal, V. D., Bushnell, M. L., "FACTS: fault coverage estimation by test vector sampling," Proc. 12th IEEE VLSI Test Symp., pp. 266-271, April 1994.

[12] Hitec/Proofs: A sequential circuit test generation and fault simulation package. The Universiry of Illinois, http://www.crhc.uiuc.edu/IGATE/hitec-software.html, updated Sept. 10, 1997. Accessed April 18, 2005.

[13] Jain, S. K., Agrawal, V. D., "Statistical fault analysis," IEEE Design & Test of Computers, vol. 2, no. 1, pp. 38-44, 1985.

[14] Lo, C. Y., Nham, H. N., Bose, A. K., "Algorithms for an advanced fault simulation system in MOTIS," IEEE Trans. CAD Integrat. Cir. & Sys., vol. 6, iss. 2, pp. 232-240, March 1987.

[15] McNamer, M. G., Roy, S. C., Nagle, H. T., "Statistical fault sampling," IEEE Trans. Indust. Elec., vol. 36, no. 2, pp. 141-150, May 1989.

[16] Seth, S. D., Agrawal, V. D., Farhat, H., "A statistical theory of digital circuit testability," IEEE Trans. Computers, vol. 39, no. 4, pp. 582-586, April 1990.

[17] Stroud, C. E., Ryan, C. A., "Multiple fault simulation with random and clustered fault injection," Proc. 8th IEEE ASIC Conf., pp. 218-221, September 1995.

[18] Thaker, P. A., Agrawal, V. D., Zaghloul, M. E., "A test evaluation technique for VLSI circuits using register-transfer level fault modeling," IEEE Trans. Computer-aided Design Integ. Circuit. Sys., vol. 22, no. 8, pp. 1104-1113, August 2003.

[19] Villoldo, J., Agrawal, P., Agrawal, V. D., "Stafan algorithms for MOS circuits," IEEE Int. Conf. Comp. Design, pp. 56-59, Oct. 1991.