Closing the Gap Between Analog and Digital

Khaled Saab
Fluence Technology Inc.
8700 SW Creekside Place
Beaverton, Or 97008, USA
(503) 672-8746
Saab@fluence.com

Naim Ben Hamida
Fluence Technology Inc.
8700 SW Creekside Place
Beaverton, Or 97008, USA
(503) 672-8725
Naim@fluence.com

Bozena Kaminska
Fluence Technology Inc.
8700 SW Creekside Place
Beaverton, Or 97008, USA
(503) 672-8739
Bozena@fluence.com

ABSTRACT

This paper presents a highly effective method for parallel hard fault simulation and test specification development. The proposed method formulates the fault simulation problem as a problem of estimating the fault value based on the distance between the output parameter distribution of the fault-free and the faulty circuit. We demonstrate the effectiveness and practicality of our proposed method by showing results on different designs. This approach extended by parametric fault testing has been implemented as an automated tools set for IC testing.

Keywords

Fault modeling, hard faults, fault simulation, test vector generation.

1. INTRODUCTION

Research in the area of analog circuit fault simulation and test vector generation has not achieved the same degree of success as its digital counterpart owing to the difficulty in modeling analog behavior, the continuous nature of their input and output signals, the non-linearity of circuit elements, and the complicated relations between input and output signal called transfer function. Thus, a direct application of digital *stuck-at* fault model proves to be inadequate for the analog circuit fault simulation and test vector generation.

Despite those difficulties, there was significant pressure from the testing community and the industry to develop a fault model and test methodology to serve the same purpose on the analog side. One major step in this direction was the migration from functional fault model to structural fault model. This move allowed fault grading and acted as a measure to quantify the quality of the test plan; thereby permitting test requirements and benchmarking of design-for-test strategies. However, analog testing was still done serially with faults inserted in the circuit and simulated one at a time.

In this paper, we present a parallel and accelerated fault simulation method that does not rely on the simulator in the loop. The proposed method builds a statistical model of the fault-free circuit and uses the circuit sensitivity and linear approximations in order to generate a fault model for each defect in the circuit. The obtained information is used for test vector specification.

Permission to make digital/hardcopy of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication and its date appear, and notice is given that copying is by permission of ACM, Inc. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
DAC 2000, Los Angeles, California
(c) 2000 ACM 1-58113-188-7/00/0006..$5.00

2. OVERVIEW

Previous work in fault simulation and test generation focuses on digital circuits using the classical *stuck-at* fault model, where serial fault simulation techniques and parallel fault simulation techniques have been developed. Serial fault simulation is the simplest method of simulating faults. It consists of transforming the fault-free circuit model *N* so that it models the circuit *Nf* created by the fault *f*. *Nf* is then simulated. The entire process is repeated for each fault of interest [3]. Other fault simulation techniques: parallel, deductive, and concurrent, have been developed and differ from the serial method in two fundamental aspects: a) they determine the behavior of the circuit *N* in the presence of fault without explicitly changing the model of *N* and b) they are capable of simultaneously simulating a set of faults.

The introduction of the *stuck-at* fault model for digital circuits enabled digital testing to cope with the exponential growth in the digital circuit size, and complexity. Indeed, the *stuck-at* fault model enabled the functional testing to be replaced by structural testing, and acted as a measure to quantify the quality of the test plan, permitting test requirements and benchmarking of design-for-test strategies.

Hard fault modeling and simulation which addresses analog and mixed-signal circuits have been a subject of many publications [1], [2], [6], [7], and [8]. In [1] it has been suggested that the faulty analog behavior should be modeled as a modification to the nominal macromodel. For example, the fault model for a transistor has been implemented by replacing each transistor with a transistor surrounded by switches as shown in Figure 1. A faulty circuit can be determined from a good one by opening or closing the appropriate switch.

In [8] to enable the circuit fault-effects to be simulated in a reasonable simulation time, behavioral models of each circuit block were developed. Hybrid fault simulations were performed by replacing each circuit block with its behavioral model equivalent except the block that has a target fault inserted. Each fault is manually inserted in the netlist for simulation. The behavioral models reduced the simulation time by a factor of 10 to 36.

All presented approaches for analog and mixed-signal circuits are all based on *cause-effect* analysis and do not allow parallel fault simulation. Indeed, *cause-effect* analysis enumerates all the possible faults (*causes*) existing in a fault model and determines all

their corresponding responses (*effects*) to a given applied test in a serial manner. The required simulation time can become impracticably long, especially for large analog designs.

The novelty of our approach consists of parallel hard fault simulation and test vector specification based on *effect-cause* analysis. Indeed, from the distance between the fault-free circuit distribution and the faulty circuit distribution (*effect*), the fault value (*cause*) for all defects could be approximated simultaneously by linear estimation. Then using fault dominance theorem and fault value data, the test vectors are derived.

The objective of this paper is to present both the methodology and a practical implementation that address two fundamental problems that are limiting the growth of the testing capabilities of analog and mixed signal circuits. To bypass those limitations, we propose a structural fault model and parallel simulation of faults. The paper is organized as follows. In the third and fourth sections, hard fault dictionary generation, fault simulation, fault coverage computation and test vector specification are presented. Implementation details, practical examples and results that demonstrate the effectiveness of the proposed methodology are included in sections four. Section five concludes the paper.

3. FAULT SIMULATION

Fault simulation is used to construct a fault dictionary. Conceptually, a fault dictionary stores the signatures of the faults for a specific stimulus T. This approach requires computing the response of every possible fault before testing, which is impractical.

We propose a new type of fault dictionary that does not store the output signature of the fault *f* (*effect*), but computes and stores the fault value R_{fault} (*cause*). When added to the circuit, this R_{fault} will drive the output parameter to go out of its tolerance box. Using this new definition of fault dictionary, two steps are needed to construct the fault dictionary:

(i) The fault list is generated, and

(ii) From the fault-free and the faulty circuit output distributions, we simultaneously compute and store the fault value R_{fault} for all defects in the fault list.

3.1 Fault Listing

The fault list contains the list of all possible shorts and opens in a circuit. Two fault list extractors could be used: a layout-based fault list (standard inductive fault analysis) and/or a schematic-based fault list extractor. In this case all branches are considered as potential opens and all nodes of the same element are considered as potential shorts. Generations of all combinations of two, three or more shorted nodes is possible but may lead to a large fault list containing a significant number of nonrealistic faults.

For MOSFET transistor, the layout extracted fault list is constructed of three shorts and two opens: short gate-source, short gate-drain, short source-drain, open drain, and open source (Figure 1).

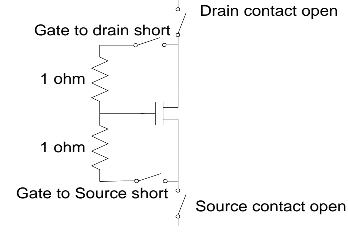

Figure 1 Transistor fault model

3.2 Fault Model (The Minimal Detectable Fault Value)

As mentioned previously, we propose to construct a fault dictionary by storing the fault value R_{fault} (*cause*) instead of the storing the output parameter (*effect*). The *effect* is a constant value which represents the detectability threshold. The *cause* is the minimal defect value that, if added to the circuit, will cause the output parameter to go out of tolerance range and makes the fault detectable.

Several methods could be used to define the detectability threshold at the output parameter:

(i) A constant absolute value deviation of the output parameter from its nominal value i.e. $D_{threshold} = 10mv$, or

(ii) A constant percentage of the output parameter value i.e. $D_{threshold} = 5\%$, or

(iii) A constant factor of the fault-free output distribution $3\sigma_{ff}$.

The first two methods of defining the detectability threshold are straightforward. For the third method (constant factor of the output distribution), we use a piecewise linear estimation to compute the fault-free circuit output due to process variations using Equation (1) [4],[5]. From Equation (1), the fault-free output mean μ_{ff} and standard deviation σ_{ff} can be obtained by Equation (2) and (3) respectively.

$$out = out_0 + \sum_{i=1}^{N} S_{x_i}^{out} \Delta x_i \tag{1}$$

$$\mu_{ff} = \mu_{out_0} + \sum_{i=1}^{N} S_{x_i}^{out} \mu_{\Delta x_i} \tag{2}$$

$$\sigma_{ff}^2 = \sum_{i=1}^{N} \left(S_{x_i}^{out}\right)^2 \sigma_{x_i}^2 + \sum_{i \neq j} \sum_{i=1}^{N} \sum_{j=1}^{N} S_{x_i}^{out} S_{x_j}^{out} \sigma_{x_{ij}} \tag{3}$$

Where out_0 is the nominal output value and out is the estimated output value due to the components variation. Δx_i is the variation of the circuit component x_i due to process variation and $S_{x_i}^{out}$ is the sensitivity of out to x_i. σ_{x_i} is the standard deviation of the component x_i, and $\sigma_{x_{ij}}$ are the covariance terms.

Now, we compute the faulty circuit nominal value μ_f and its standard deviation σ_f due to added resistance. Equation (1), (2), and (3) respectively become

$$out = out_0 + \sum_{i=1}^{N} S^{out}_{x_i}\Delta x_i + G^{out}_{f_i} R_{f_i} \quad (4)$$

$$\mu_f = \mu_{out_0} + \sum_{i=1}^{N} S^{out}_{x_i}\mu_{\Delta x_i} + G^{out}_{f_i}\mu_{R_{f_i}} \quad (5)$$

$$\sigma_f^2 = \sum_{i=1}^{N}\left(S^{out}_{x_i}\right)^2\sigma_{x_i}^2 + \sum_{i\neq j i=1}^{N}\sum_{j=1}^{N} S^{out}_{x_i}S^{out}_{x_j}\sigma_{x_{ij}} + \left(G^{out}_{f_i}\right)^2\sigma^2_{R_{f_i}} + \sum_{i=1, j=cte}^{N} G^{out}_{f_j}S^{out}_{x_i}\sigma_{x_iR_{f_j}} \quad (6)$$

Where R_{f_i} is the newly added component due to a short or open and $G^{out}_{f_i}$ is the gradient of out with respect to the fault f_i. $\sigma_{R_{f_i}}$ is the standard deviation of the resistance, and $\sigma_{x_iR_{f_j}}$ the covariance term between the newly added component and the components in the original circuit.

Since $\sigma_{R_{f_i}}$ is always zero, and if we consider the variable as independent, the covariance terms $\sigma_{x_iR_{f_j}}$ are zero, the expression for the faulty output variance is greatly simplified, and Equation (6) reduces to

$$\sigma_f^2 = \sum_{i=1}^{N}\left(S^{out}_{x_i}\right)^2\sigma_{x_i}^2 + \sum_{i\neq j i=1}^{N}\sum_{j=1}^{N} S^{out}_{x_i}S^{out}_{x_j}\sigma_{x_{ij}} \quad (7)$$

Thus, under the above assumptions, hard defects do not modify the fault-free circuit standard deviation but affect only the mean value and $3\sigma_f = 3\sigma_{ff}$.

Now that the fault-free and faulty output distributions are obtained, the detectability threshold is set to be the minimal distance between the fault-free and faulty circuit that guarantee detectability of the fault (Figure 2)

$$D_{threshold} = \mu_f - \mu_{ff} \approx 3\sigma_f + 3\sigma_{ff} \quad (8)$$

where μ_{ff} and μ_f are the mean values for the fault-free and faulty circuit and σ_{ff} and σ_f are the estimated fault-free and faulty output standard deviations.

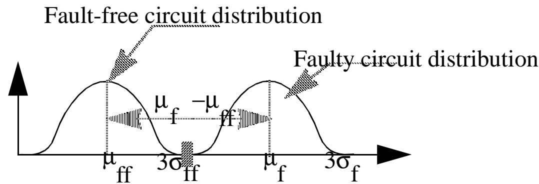

Figure 2 Fault-free and faulty circuit output

The fault value R_{f_i} could now be obtained from Equation (2) and (5):

$$R_{f_i} \approx \frac{|\mu_f - \mu_{ff}|}{G^{out}_{f_i}} \approx \frac{|3\sigma_f + 3\sigma_{ff}|}{G^{out}_{f_i}} \quad (9)$$

$$R_{f_i} \approx \frac{6\sigma_{ff}}{G^{out}_{f_i}} \quad (10)$$

Combining (3) and (10), we obtain

$$R_{f_i} \approx \frac{6\sqrt{\sum_{i=1}^{N}\left(S^{out}_{x_i}\right)^2\sigma_{x_i}^2 + \sum_{i\neq j i=1}^{N}\sum_{j=1}^{N} S^{out}_{x_i}S^{out}_{x_j}\sigma_{x_{ij}}}}{G^{out}_{f_i}} \quad (11)$$

S and G are obtained using the well known adjoint network method [10]-[12].The adjoint network method allows us to compute the sensitivity of one output parameter with respect to all component variations (existing and non-existing components) in only two simulations, one for the original network and one for the corresponding adjoint network. The adjoint network method for sensitivity computation in AC, DC and transient domain has been implemented using Hspice [13] as a basic simulator [4],[5] and [9].

In summary, the algorithm consists of two steps. First, from fault free circuit simulation, we compute the fault-free mean and standard deviation of the output parameters due to process variations [5]. Then, in the second step, from the output parameter distributions and gradient values, we compute and store the resistance values that will drive the output parameter(s) out of their tolerances (Figure 3). Note that the obtained resistance value indicates the value of the resistance that, if added to the branch (open circuit case) or between two nodes (short circuit case), will cause the output parameter to go out of its tolerance range. This resistance value will be used for test vector specification.

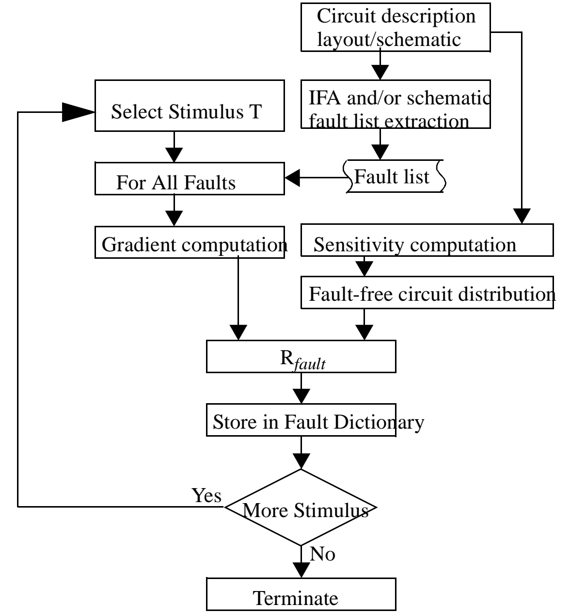

Figure 3 Fault Dictionary Construction

4. TEST VECTOR GENERATION

In this section we describe an algorithm which uses the fault dictionary generation approach proposed earlier and the fault dominance concept to derive the fault coverage and the set of test vectors that detect the largest set of faults without targeting individual faults.

4.1 Fault Dominance

In digital circuits, fault dominance is used to reduce the number of faults that need to be considered.

Definition [3]: Let T_g be the set of all tests that detect a fault g. We say that a fault f dominates the fault g if f and g are equivalent under T_g.

In other words, if f dominates g, then any test that detects g, will also detect f (on the same primary output). Therefore, for fault detection it is unnecessary to consider the dominating fault f, since by deriving a test to detect g we automatically obtain a test that detects f as well.

In analog circuits, the input output relationship is more complex, but to the first order approximation, the fault dominance theorem could be used as well.

Indeed, instead of testing for the upper and lower resistance values for faults as in [8] (Table 1), in our case, fault dominance will be used where only the least dominating resistance (the largest) will be tested for shorts, and the least denominating resistance (the smallest) will be tested for open circuits.

TABLE 1 Upper and lower resistances used for hard fault modeling.

Defect Type	Lower Resistance (ohms)	Upper Resistance (ohms)
Added metal 1	0.2	1000
Added metal 2	0.2	1000
Via short	5	5
Junction leakage	100	10 000
Poly-metal 1 short	0.2	1000
Poly-metal 2 short	0.2	1000
Poly-poly short	20	1000
Open	1Meg	∞

4.2 Test Vector Specification

What we need now is to determine, for each fault, what is the best stimulus for detecting this fault. This best stimulus will be considered as the test vector for this fault.

The algorithm for performing this selection (Figure 4) consists of the following steps: a) select a set of stimuli. A default stimuli (user defined) could be used or it can be made by the test engineers in an interactive mode in order to consider any special characteristic of the circuit. Stimuli are divided into DC, AC and transient stimulus [2]: sine wave, pulse, ramp, any PWL function, etc.; b) construct the fault dictionary and obtain R_{fault} as a function of stimulus T; c) loop through all stimulus and all faults in the fault dictionary and compare the fault value R_{fault} with the typical values for short and opens $R_{typical}$.(Table 1); d) evaluate the fault observability for each one of the proposed stimulus; and e) select the stimulus for which the fault observability is maximal and mark it as the test vector for the fault under consideration.

The technique for open circuit faults will be described in the following section. The exact same reasoning is used for short circuit faults.

Knowing the component tolerance Δx_i, the output parameter distribution of the fault-free and faulty circuit $3\sigma_{ff}$ and $3\sigma_f$ can be estimated. Then from equation (11) we estimate the resistor value R_{fault} that will cause the output parameter to go out of the tolerance range. R_{fault} are stored in fault dictionary.

Table 1 presents a list of circuit defects and the corresponding typical resistance range obtained by statistical analysis of circuit defects [8]. R_{fault} is compared to $R_{typical}$. If R_{fault} is less than $R_{typical}$, we conclude that R_{fault} (or a higher resistance value) *will cause the output parameter to deviate out of tolerance, and the stimulus T is* ***accepted*** *as a valid test vector.* On the other hand, if R_{fault} is higher than $R_{typical}$, no deviation of the output parameter could be detected and the stimulus T is **rejected**.

Finally, from the set of stimulus that has been accepted as valid test vectors, select the stimulus for which R_{fault} is minimal (the dominating fault value) and consider it as the test vector for the current fault.

Since shorts are the dual of opens, we do not need to repeat the steps above where the same technique is applied, except that R_{fault} needs to be higher than $R_{typical}$ in order to cause a detectable deviation in the output parameter signature.

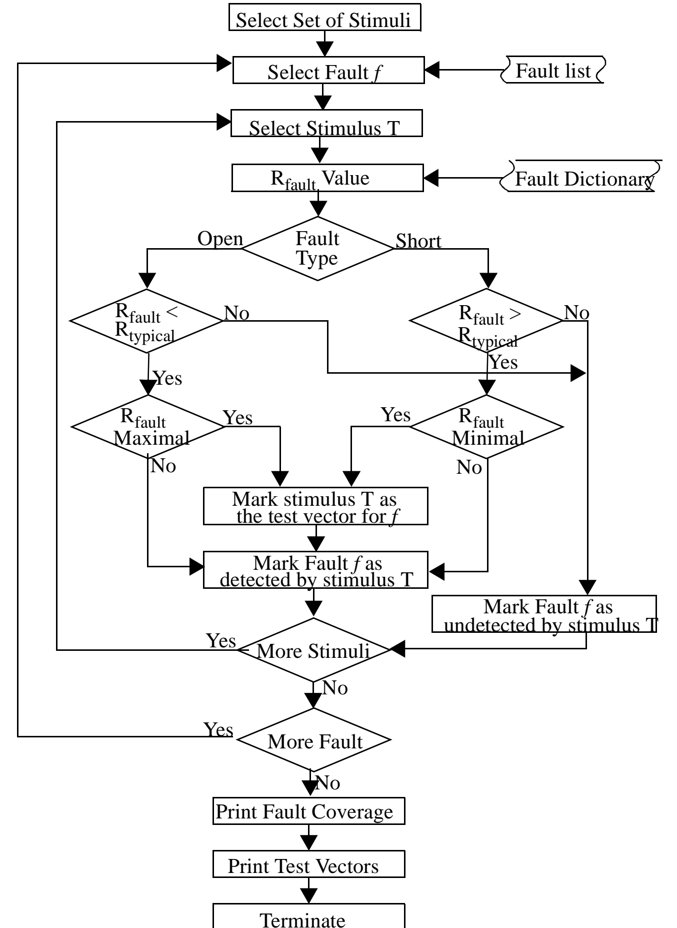

Figure 4 Test Vector Generation Flow

5. EXPERIMENTAL RESULTS

The proposed algorithm has been implemented as an automated tool set for IC testing, fault simulation and test vector generation. The implementation was done in the C programming language on a SUN workstation.

The procedures outlined in this paper were applied to a fifth order Chebychev band-pass filter (Figure 5), and to several benchmark circuits as shown in Table 4.

Detailed example:

The fifth order Chebychev filter (Figure 5), is tested for 25 possible hard faults. The stimulus set T was selected as the frequency range [0,20KHz]. The detection threshold was set to 5% of the nominal output voltage, $R_{typical}$ for open set to 1Meg ohm, and $R_{typical}$ for shorts set to 1Kohm.

It is found that all the relevant possible shorts and opens are easily detected. Table 2 and Table 3 give some possible hard faults, their computed nominal values and the frequency/test vector at which they are detected.

.

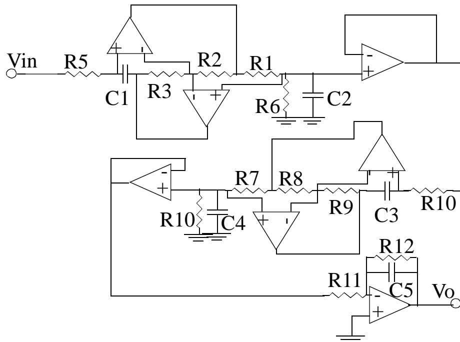

Figure 5 Fifth Order Chebychev Filter

TABLE 2 Fifth Order Chebychev Filter, Open Circuit Fault Results

OPEN	R_{fault} [ohm]	$R_{typical}$ [ohm]	Frequency [Hz]	Decision
R1	1.1k	1Meg	300	Accepted
R2	1.93k	1Meg	300	Accepted
R3	400	1Meg	0.1	Accepted
R4	662.5	1Meg	0.1	Accepted
R5	4.83k	1Meg	100	Accepted
R6	2.78k	1Meg	200	Accepted
R7	2.78k	1Meg	200	Accepted
R8	431	1Meg	0.1	Accepted
R9	555	1Meg	0.1	Accepted
C1	5.12K	1Meg	100	Accepted

TABLE 2 Fifth Order Chebychev Filter, Open Circuit Fault Results

OPEN	R_{fault} [ohm]	$R_{typical}$ [ohm]	Frequency [Hz]	Decision
C2	4.43k	1Meg	100	Accepted
C3	2.89k	1Meg	100	Accepted
gm1	872	1Meg	200	Accepted
gm2	243	1Meg	0.1	Accepted
gm3	1k	1Meg	1k	Accepted

TABLE 3 Fifth Order Chebychev Filter, Short Circuit Fault Results

SHORT	R_{fault} [ohm]	$R_{typical}$ [ohm]	Frequency [Hz]	Decision
Vin & 0	INF	1k	all	Accepted
VDD & 0	INF	1k	all	Accepted
VSS & 0	INF	1k	all	Accepted
R1 & 0	20Meg	1k	0.1	Accepted
R1 & R2	8Meg	1k	0.1	Accepted
R5 & R6	85k	1k	0.1	Accepted
R9& 0	50M	1k	1k	Accepted
in-&0	25Meg	1k	0.1	Accepted
in+ & 0	2e-5	1k	0.1	Rejected

The obtained fault coverage is 96% with only 5 test vectors (0.1Hz, 100Hz, 200Hz, 300Hz, 1KHz).

Experimental results (summary):

A set of 7 benchmark circuits are simulated. The benchmark circuits range from a simple operational amplifier to a complex 8 bit current DAC. Level 3, level 28 and level 49 transistor MOSFET models are used in the sensitivity computation environment.

In all experiments the output voltage measurements and/or output current measurements were considered for testing. The detection threshold was set to 6σ where σ is the standard deviation of the fault-free circuit. Table 4 describes the circuit type, the test domain, the number of transistors in the circuit, the number of faults in the fault list obtained by a schematic fault dictionary extractor, and the total CPU time on a SPARC 10 workstation. The approximated CPU time is given for serial methods using one simulation per fault.The obtained fault coverage is also provided for comparison purposes.

TABLE 4 Benchmark Results

Circuit	Domain	# of transistors	# of faults	Test-Maxx CPU time [sec]	Serial method CPU time [sec]*	Fault coverage [%]
Inverter	DC	9	47	0.28	5.68	91.5
low pass filter	Trans	9	51	0.43	7.92	94.1
State variable filter	AC	36	184	3.64	334.88	85.3
Chebychev filter	AC	27	149	5.32	348.16	87.2
4 bit ADC	DC	153	737	0.19k	36.68k	85.7
8 bit ADC flash	DC	63	295	0.02k	1.66k	87.4
8 bit current DAC	Trans	132	669	1.20k	267.6k	98.6

*estimated based on one simulation per fault

The average fault coverage was 90% with a small CPU cost. Figure 6 presents the fault coverage as a function of the number of test vectors.

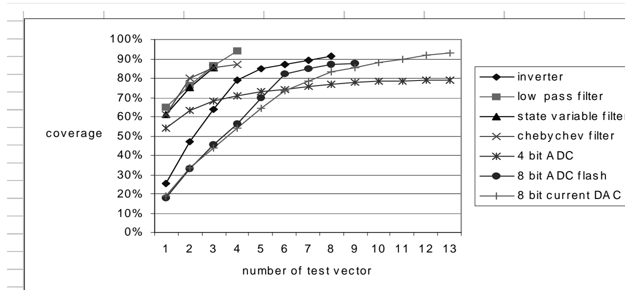

Figure 6 Fault Coverage as Function of the Number of Test Vectors

6. CONCLUSION

By using a structural fault model and by moving from the standard serial test approaches to a parallel test approach, mixed-signal circuit testing is now one step closer to the digital testing. Indeed, an automatic tool for parallel fault simulation and test vector generation based on cause-effect approach has been presented. The concept is new, and from our simulation results, we can conclude that this method is highly efficient. Several examples and test cases showed that fault coverage was as high as 98.6% with simulating time several order of magnitude less than the serial methods.

7. ACKNOWLEDGMENTS

We wish to thank Jim Abbate for his help in generating the results.

8. REFERENCES

[1] L. Milor and V. Visvanathan, “Detection of Catastrophic Faults in Analog Integrated Circuits”, IEEE Trans. Computer-Aided Design, Vol. CAD-8, no. 2, pp. 114-130, Feb. 1989.

[2] Manoj Sachdev, “A Realistic Defect Oriented Testability Methodology For Analog Circuits”, JETTA 1993.

[3] M. Abramovici, M.A. Breuer, A.D. Friedman, “Digital Systems Testing and Testable Design”, IEEE press, 1990.

[4] Naim B-Hamida, Khaled Saab, David Marche and Bozena Kaminska,”FaultMaxx: A Perturbation Based Fault Modeling and Simulation for Mixed-Signal Circuits”, Asien Test Conference. October 1997. pp. 182-187

[5] Naim B. Hamida, Khaled. Saab, David. Marche, B. Kaminska and Gay. Quesnel, “LIMSoft: Automated Tool for Design and Test Integration of Analog Circuits”, International Test Conference 1996. pp. 56-71

[6] Naveena Nagi and Jacob A. Abraham, “Hierarchical Fault Modeling For Analog and Mixed-Signal Circuits”, IEEE VLSI Test Symposium 1992, pp92-101.

[7] Naveena Nagi, A. Chatterjee, Jacob A. Abraham, “Fault Simulation Of Linear Analog Circuits”, Analog Integrated Circuits and Signal Processing 1993.

[8] R. J. A. Harvey, A. M. D. Richardson, E, M. J. G. Bruls, K. Baker, “Analog Fault Simulation Based on Layout Dependent Fault Models”, ITC 95, pp-641-649.

[9] Saab Khaled, Naim B. Hamida, David Marche and Bozena Kaminska, “LIMSoft: Automated Tool for Sensitivity Analysis and Test Vector Generation”, IEE Proceedings on Circuits, Devices and Systems. December 1996. pp.386-392

[10] Stephan W. Director and Ronald A. Rohrer, “Automated Network Design- The Frequency Domain Case”, IEEE Trans. on Circuit Theory, CT-16, no 3, August 1969, pp. 330-337.

[11] Stephan W. Director and Ronald A. Rohrer, “The Generalized Adjoint Network and Network Sensitivities”, IEEE Trans. on Circuit and Theory, Vol. CT-16, no 3, August 1969, pp. 318-323.

[12] Tellegen B. D. H., “A General Network Theorem, with Application”, Phillips Res. Dept., no. 7, pp. 259-269.

[13] Meta-Software, “HSPICE User’s Manual version H92”, Meta-Software, Inc. 1992.