Cost-Effective Concurrent Test Hardware Design for Linear Analog Circuits*

Sule Ozev and Alex Orailoglu
University of California, San Diego
La Jolla, CA 92093
{sozev, alex}@cs.ucsd.edu

Abstract

Concurrent detection of failures in analog circuits is becoming increasingly more important as safety-critical systems become more widespread. A methodology for the automatic design of concurrent failure detection circuitry for linear analog systems is discussed in this paper. In contrast to previous approaches, the methodology aims at providing coverage in terms of all the circuit components while minimizing the loading overhead by reducing the number of internal circuit nodes that need to be tapped. Parameter tolerances are incorporated through either statistical or mathematical analysis to determine the threshold for failure alarm. Experimental results confirm that full coverage can be attained while keeping the hardware overhead within a pre-specified budget.

1. Introduction

With feature sizes rapidly approaching nanometer scales, electronic circuits are ever more susceptible to external upsets, such as electron migration and radiation, which result in permanent changes in circuit component values, eventually causing system failures. As safety critical systems find more widespread use in today's high technology era, concurrent detection of possible system failures is all the more important as the susceptibility of electronic circuits to environmental effects constantly increases.

Most electronic systems interact with the outside world through analog front and back end subsystems. Concurrent detection of failures in analog circuits is essential to ensuring correct operation since the analog subsystems, such as signal capturing devices, are most likely to be placed on system boundaries, heightening their exposure to environmental effects.

Whereas numerous methodologies have been developed for designing on-line test circuits in the digital domain [4, 1], a limited set of approaches only exists in the analog domain [2, 3, 5].

State-variable representations of linear analog circuits are utilized in [2, 3]. The system invariant is defined by multiplying the state space matrix with a row vector that corresponds to the weights associated to each equation. This invariant is then implemented by observing the state variables. A number of drawbacks conspire against the practical applicability of this approach. Perhaps the most problematic is the requirement for direct observation of *all* state variables. The inability of the proposed method despite considerable area overhead to deliver full coverage enables test escapes; the problem of escapes is accentuated by the inability to handle the important challenge in analog test of parameter tolerances. High rates of false alarms resulting from process variations may interfere with the fault-free operation of the circuit, furthermore.

Another state-space representation based approach to designing on-line detection hardware for analog circuits is presented in [5]. This approach aims at elimination of the invariant's dependency on the state variables that cannot be easily observed. Rather, the invariant is made to depend on observable node voltages. The authors also aim at optimizing the detection circuit by minimizing its sensitivity to noise generating components, such as resistors and maximizing the sensitivity to other components. Whereas the hardware overhead depends on the elimination of state variables from the system of equations, full coverage of all the circuit components cannot be guaranteed as their effect on the invariant can be nullified during elimination of state variables from the system of equations.

Coverage is the most important issue for concurrent test. The detection circuit needs to be sensitive to catastrophic failures in *all* the circuit components in order to detect system failures. On the other hand, small variations in circuit component values should not cause an alarm since such parametric variations are an inevitable aspect of the manufacturing process and tolerated in the normal operation mode in the analog domain.

In this paper, we present a methodology for designing

* This work is supported through an IBM fellowship, the National Semiconductor Corp., and the UC Micro Office.

1063-6404/02 $17.00 © 2002 **IEEE**

on-line test hardware applicable to linear analog circuits. In most previously proposed analog on-line test approaches, the detection circuit cannot guarantee coverage in terms of all the circuit components [3, 5]. In addition, the complexity of the resulting hardware overhead is unpredictable. In contrast to previously proposed approaches, the methodology proposed herein delivers solutions that provide 100% fault coverage while remaining within a specified hardware budget and minimizing the number of circuit nodes to be tapped.

2. Methodology

The goal of an on-line test methodology is to find an invariant in the system that can easily be checked by hardware. In linear analog systems, the invariants are the circuit components, such as resistors, capacitors, and inductors. However, it will be impractical to check the soundness of each component individually due to extreme hardware overhead. Thus, a methodology to combine these components is needed so that the resulting invariant can be implemented within acceptable hardware overhead. It is also important that the size of the hardware circuitry does not blow up rapidly with increasing circuit size.

Since the goal of concurrent testing is to detect failures in any circuit component, a viable approach needs to incorporate the component coverages into the design flow of the detection circuitry. The variation in circuit parameters obviates the possibility of checking the output of the detection circuit against a single value, instead forcing reliance on threshold violations for alarming a failure. In order to prevent false alarms, this threshold needs to be determined by utilizing statistical information about process variations. Hardware overhead and loading of internal circuit nodes are important issues in designing such concurrent test circuitry.

Nodal equations define relations between the node voltages in a circuit based on the values of circuit components. In linear circuits, the relations between the internal node voltages do not depend on the input waveform and therefore constitute an invariant for the system.

As a result, the nodal equations can be utilized to define and implement an invariant that can be checked on-line. In order to limit the performance overhead of the detection circuitry, one has to limit the number of internal points that need to be observed. Moreover, some circuit nodes, such as the terminals of the internal resistance of a capacitor, may not even be physically accessible. Therefore, the detection circuitry needs to be designed with a subset of circuit nodes and with as little hardware overhead as possible. It is also necessary to predict the coverage of the detection circuit in terms of circuit components during the design process.

The proposed methodology aims at the automatic design of an on-line test circuitry by combining nodal equations in order to reduce the number of nodes that need to be observed while keeping the hardware overhead within acceptable bounds.

We start with a given set of observable internal nodes in the circuit. The nodal equations for this set of nodes can be derived by expressing the voltages of unobservable nodes in terms of the observable nodes. The resulting differential equation system is in the form of:

$$(A_n s^n + ... + A_0)V + (B_m s^m + ...B_0)V_i = 0$$

where V is the vector of the observable nodes and V_i is the input vector. The larger of (n, m) defines the inherent degree of the system with the given set of observable nodes. The minimal hardware overhead of the detection circuitry depends on this inherent degree of the system as the number of integrators required to implement a combination of these equations is equal to the degree of the equation system.

The initial set of observable nodes may still introduce too much performance overhead as each tapping point loads and injects noise into the system. In most cases, there is a set of nodes that can be deleted from the observation set with no increase in the degree of equations, and thus no increase in detection hardware. These correspond to the nodes that involve no differentiation in the nodal equations. Therefore, identification of such nodes and their deletion from the system of equations is the first step in minimizing the number of observable nodes. A number of circuit nodes can still be deleted from the system of equations at the cost of increasing the degree of the system, thus increasing the detection hardware overhead. If the hardware overhead is within the given initial budget, the next step is to select a set of nodes to delete from the observable set with minimal increase in the degree of the system. This step can be repeated until the hardware budget is reached or until there are only two nodes left in the observable node set.

The next phase is to combine a set of equations in the resulting system such that all the circuit components appear in at least one equation. The left side of the equations can simply be added as the right sides are all zeros. The resulting sum then involves differentiations of the circuit node voltages. However, differentiating node voltages may cause instability in the circuit, and therefore is undesirable. Fortunately, since the right sides of the equations are all zero, the sum of the left sides can be integrated k times, k being the degree of the system of equations after the application of the methodology. This final integrated sum yields the form of the detection circuitry.

Once the invariant is determined in terms of a combination of equations, it needs to be checked against a threshold to alarm a failure. This threshold mainly stems from the variations in circuit parameters and it can be determined through the utilization of either circuit simulations or mathematical analysis.

3. Minimizing the Number of Tapped Nodes

In order to reduce the performance overhead of the detection circuit due to loading and noise, the number of tapped nodes needs to be reduced. In some cases, a set of nodes may not be involved in any differentiation, enabling their voltage to be expressed as a linear combination of other node voltages. Elimination of such nodes from the system of equations affects neither the degree of the system of equations nor the hardware overhead of the detection circuit, consequently. However, deletion of the nodes that *are* involved in differentiation results in an increase in the degree of the system, and a consequent increase in the hardware overhead. The level of such an increase depends on the nature of the system. By analyzing the system matrices, the cost of eliminating each node can be computed and this information can be utilized in selecting the nodes to be dropped from the observation set.

3.1. No Cost Node Elimination

The first step in minimizing the number of observable nodes is to identify the nodes that do not involve any differentiation. By expressing the voltages of these nodes in terms of the rest of the nodes in the system, the degree of the system is kept intact. The number of tapped nodes is consequently reduced at no increase in the hardware overhead. The initial system of equations is in the form of:

$$(A_n s^n + ... + A_0)V + (B_m s^m + ...B_0)V_i = 0$$

The set of nodes that can be deleted with no hardware overhead have corresponding zero columns in matrices A_n through A_1. By identifying these columns, we have:

$$\{\begin{bmatrix} A_n^i & 0 \end{bmatrix} s^n + ... + \begin{bmatrix} A_0^i & A_0^{ii} \end{bmatrix}\} \begin{bmatrix} V^i \\ V^{ii} \end{bmatrix} + \{\begin{bmatrix} B_m \end{bmatrix} s^m + ... + \begin{bmatrix} B_0 \end{bmatrix}\} V_i = 0$$

In order to avoid any increase in the degree of the system, the nodes to be deleted have to be expressed as a linear combination of the rest of the nodes. The set of nodes to be deleted and the associated linear combinations to be used can be captured in the maximal nonsingular submatrix, A_0^{vi}, of A_0^{ii}. The structure of this submatrix and its position within the general matrix formulation can be shown through a horizontal partitioning of the previous matrix formulation.

$$\{\begin{bmatrix} A_n^{iii} & 0 \\ A_n^{iv} & 0 \end{bmatrix} s^n + ... + \begin{bmatrix} A_0^{iii} & A_0^{iv} \\ A_0^{v} & A_0^{vi} \end{bmatrix}\} \begin{bmatrix} V^{iii} \\ V^{iv} \end{bmatrix}$$

$$+\{\begin{bmatrix} B_m^{iii} \\ B_m^{iv} \end{bmatrix} s^m + ... + \begin{bmatrix} B_0^{iii} \\ B_0^{iv} \end{bmatrix}\} V_i = 0$$

The nodes corresponding to the columns of A_0^{ii} can easily be deleted from the system by utilizing equations corresponding to the rows of A_0^{ii}:

$$V^{iv} = -A_0^{vi^{-1}}\{A_n^{iv} s^n + .. + A_0^{iv}\}V^{iii} - A_0^{vi^{-1}}\{B_m^{iii} s^m + .. + B_0^{iii}\}V_i$$

Substituting V^{iv} into the remaining equations, we obtain:

$$\{(A_n^{iii} - A_0^{vi^{-1}} A_n^{iv})s^n + ... + (A_0^{iii} - A_0^{vi^{-1}} A_0^{iV})\}V^{iii} +$$

$$\{(B_m^{iii} - A_0^{vi^{-1}} B_m^{iv})s^m + ... + (B_0^{iii} - A_0^{vi^{-1}} B_0^{iv})\}V_i = 0$$

3.2. Tradeoffs in Node Elimination

Further decreasing the number of tapped nodes necessitates an increase in the degree of the system. In order to determine which nodes can be deleted without exceeding the specified hardware budget, the deletion cost of each node needs to be identified. Deleting a node necessitates expressing its voltage in terms of the rest of the node voltages. Whereas one can utilize any equation that the voltage of the node to be deleted appears in, this may lead to poles in the resulting expressions, undesirable since for each pole, a different integrator needs to be implemented. Use of an equation in which the targeted node is not involved in any differentiation saves the additional cost. The deletion of a node thus depends on the existence of at least one equation, in which the corresponding entries in matrices A_n through A_1 are zero. The next step in the method is to identify a set of equations that satisfy this condition. The consequent increase in the degree of the system for deleting a node utilizing an equation is the sum of the degree of the equation and the degree of the node. The cost of deleting that node is therefore the minimum increase in the system degree. After computing the cost of deleting each node, a candidate set of nodes and equations is selected such that deletion of the nodes utilizing the equations leads to the minimal increase in system degree. By re-arranging the system of equations, with respect to the cost information, we obtain:

$$\{\begin{bmatrix} A_n^k & A_n^{k+1} \\ A_n^{k+2} & 0 \end{bmatrix} s^n + ... + \begin{bmatrix} A_0^k & A_0^{k+1} \\ A_0^{k+2} & A_0^{k+3} \end{bmatrix}\} \begin{bmatrix} V^k \\ V^{k+1} \end{bmatrix}$$

$$+\{\begin{bmatrix} B_m^k \\ B_m^{k+1} \end{bmatrix} s^m + ... + \{\begin{bmatrix} B_0^k \\ B_0^{k+1} \end{bmatrix}\} V_i = 0$$

where columns of A_0^{k+3} correspond to the nodes that can be deleted utilizing the equations corresponding to the rows of A_0^{k+3}. The submatrix, A_0^{k+3}, is not necessarily square as there may be more than one equation to utilize for deleting each node. The number of columns of A_0^{k+3} determines the maximum number of nodes that can be deleted at once.

However, as the voltages of the nodes are interrelated in the system of equations, attempting to delete all the nodes in the candidate set may increase the degree of the system beyond the predicted minimum. The largest non-singular matrix in A_0^{k+3} determines which of the identified nodes can be deleted at once without increasing the hardware overhead beyond the computed minimum. Therefore, it is essential to find the largest nonsingular submatrix within A_0^{k+3} in order to provide the largest reduction in the number of tapped circuit nodes. The set of nodes that can be deleted simultaneously, V^{k+5}, can be determined through:

$$V^{k+5} = -(A_0^{k+7})^{-1}\{A_n^{k+6}s^n + ... + A_0^{k+6}\}V^{k+4}$$
$$-(A_0^{k+7})^{-1}\{B_m^{k+7}s^m + ... + B_0^{k+7}\}V_i$$

where A_0^{k+7} is the largest non-singular submatrix of A_0^{k+3}, V^{k+5} is the subset of V^{k+1} corresponding to A_0^{k+7}, and V^{k+4} is the remaining set of node voltages.

This step can be repeated until the hardware budget is reached or until the number of tapped nodes is reduced to two, corresponding to the input and output nodes.

4. Selection and Implementation of Equations

While in the system of equations, a component may appear in more than one equation, in the final detection circuit, it is sufficient to include a single occurence. Subsequent to the minimization of the number of tapped nodes, a set of the remaining equations that provide the desired level of coverage can be selected for the final detection circuitry. The benefit associated with each equation is simply the number of components it covers and the cost of each equation is its degree. In order to select the equations to be implemented, we start with components that are covered by the smallest number of equations. The equations that cover most of these components are selected first. If there exist more than one equation that satisfies this condition, the equation that includes the lowest number of already covered components is selected so as to even out the sensitivity of the invariant to all components. The algorithm is iterated on the remaining set of components until all the components are covered by at least one equation. When the final set of equations to be implemented is determined, they can be composed into an invariant by multiplying both sides with a row vector:

$$[\omega_1...\omega_j]\{(A_p s^p + ... + A_0)V + (B_q s^q + ... + B_0)V_i\} = 0$$

The column vector $\Omega = [\omega_1...\omega_j]$ corresponds to the weights assigned to each equation, and $max(p,q)$ denotes the degree of the invariant. The goal in introducing the weights is to reduce the distance between the invariant's sensitivities to each circuit component. In this way, each component will be evenly covered. In the proposed method, a simplified notion of sensitivity (S_k^i) is utilized by computing the effect of each component on an equation as the ratio of the number of times the component in question appears to the number of times the remaining components appear in that equation.

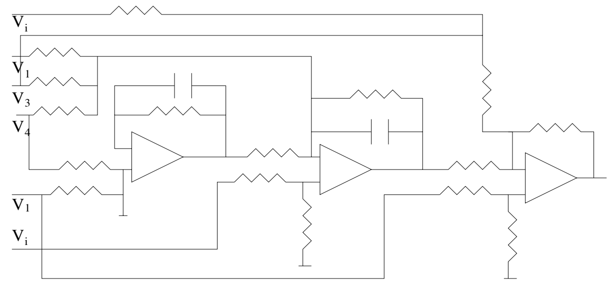

Figure 1. Second Order Detection Circuit

$$S_k^i = \frac{\#\text{ of times Component } k \text{ appears in Equation } i}{\#\text{ of times Component } j \text{ appears in Equation } i} \quad j \neq k$$

In a physical implementation of the detection circuit, the resulting invariant needs to be low-pass filtered to eliminate differentiations since they introduce instabilities. The degree of the filter has to be at least as large as the degree of the invariant (max(p, q)) in order to achieve a stable design. Since increasing the degree of the filter beyond this value will only result in additional hardware cost, the low-pass filter chosen has the same degree as the invariant. This filtering is done by introducing poles $s+\tau_n$ through $s+\tau_1$:

$$\mathcal{I} = \frac{\sum_{k=1}^{N} H_k(s)V_k + \sum_{k=1}^{M} H_k(s)V_{i_k}}{\prod_{k=1}^{r}(s+\tau_k)} \quad (1)$$

where H_k corresponds to the coefficient of node or input voltage, N is the number of tapped nodes, M is the number of inputs, and r is the degree of the final system of equations ($r = max(p,q)$).

As an example, a detection circuit of second degree with three tapped nodes, (V_1, V_3, V_4), is shown in Figure 1.

4.1. Determination of Threshold

Due to the fluctuations in the manufacturing process, circuit components are allowed to have values that vary within a certain tolerance. The system of equations that characterizes the circuit is derived by utilizing the nominal values of the parameters. The designed invariant then has a nominal value of zero. However, the difference between the actual and nominal values of circuit components will cause the invariant to be nonzero. Since some fluctuation in circuit components needs to be allowed, checking the invariant against zero introduces a high percentage of false alarms. Therefore, a threshold on the invariant has to be determined to minimize false alarms.

Circuit parameters exhibit a gaussian-like probability distribution around their nominal value. The invariant will

Expression	Mean	Standard Deviation
$z = x + y$	$\mu_z = \mu_x + \mu_y$	$\sigma_z = \sqrt{\sigma_x^2 + \sigma_y^2}$
$z = xy$	$\mu_z = \mu_x \mu_y$	$\sigma_z = \sqrt{\sigma_x^2 \sigma_y^2 + \mu_x^2 \sigma_y^2 + \mu_y^2 \sigma_x^2}$
$z = \frac{1}{x}$	$\mu_z = \frac{1}{\mu_x}$	$\sigma_z = \frac{\sigma_x}{\mu_x^2}$

Table 1. Computing Composite Distributions

also have a similar probability distribution and the threshold for raising a failure flag can be adjusted to trade off the rate of false alarms with fault coverage. The problem then reduces to computing this probability distribution of the invariant, utilizing the probability distributions of the circuit components.

The most straightforward solution in determining the probability distribution of the invariant is utilizing statistical sampling methods. Monte-Carlo analysis is the most widely used statistical sampling technique utilized in circuit design. The simplicity of Monte-Carlo simulations makes them advantageous to utilize. However, if better accuracy is desired, deterministic statistical sampling methods, such as Taguchi's method [6], can be used.

For large circuits, use of statistical sampling methods may introduce high computation times. An alternative solution to determine the threshold is to approximate the distribution of the invariant by utilizing its analytical expression. In order to enable this analysis, the invariant has to be expressed in terms of the independent input voltages. Thus, the node voltages in (1) have to be eliminated from this expression. This can be easily achieved using the transfer function of the node voltages:

$$V = H(s)V_i; \qquad \mathcal{I} = \frac{\sum_{k=1}^{M} H_k(s) V_{i_k}}{\prod_{k=1}^{r} (s + \tau_k)} \tag{2}$$

The goal is now to estimate the distribution of the invariant in the frequency domain in terms of the distributions of the circuit components using the analytical expressions given in Table 1. The maximum 3σ point of the invariant throughout the frequency spectrum determines the threshold.

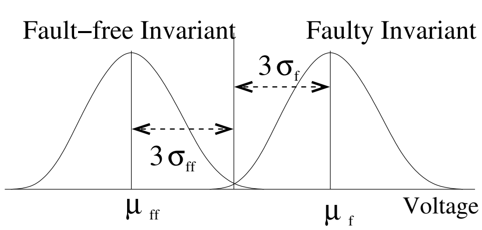

Figure 2. Failure Alarm

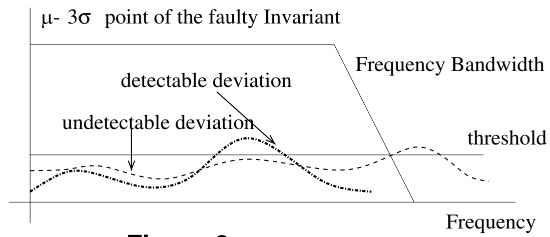

Figure 3. Detection Criterion

4.2. Fault Coverage

Since circuit components are continuous, it is not possible to come up with a single fault coverage number. In the proposed method, the quality of the on-line test circuit is assessed by computing the minimum deviation that can be detected for each circuit component.

In order for a certain level of component deviation to be classified as *detected*, it has to result in an invariant response above the assigned threshold. The invariant response under a component deviation once again exhibits a gaussian-like distribution. If the distribution invariant for with and without the deviation are disjoint, that deviation can be detected with the on-line test circuit. In other words, the lower 3σ point of the invariant with the assigned deviation has to be higher than the computed threshold as shown in Figure 2. In order to compute the minimum detectable deviations, the application operating range of the circuit needs to be considered; this detection criterion is illustrated in Figure 3.

5. Experimental Results

Utilizing the presented method, an on-line detection circuit has been constructed for a simple RC circuit given in Figure 4.

Initially, all nodes are assumed to be observable. The hardware budget of the on-line detection circuitry is assigned to be of second degree. The initial system of equations obtained through modified nodal analysis is of first degree. It can be written as:

$$\left\{ \begin{bmatrix} C_2 & 0 & -C_2 & 0 & 0 \\ -C_1 & C_1 & 0 & 0 & 0 \\ -C_2 & 0 & C_2+C_3 & 0 & -C_3 \\ 0 & 0 & 0 & C_4 & 0 \\ 0 & 0 & -C_3 & 0 & C_3 \end{bmatrix} s + \begin{bmatrix} G_T & -G_2 & -G_3 & 0 & 0 \\ -G_2 & G_2 & 0 & 0 & 0 \\ -G_3 & 0 & G_3 & 0 & 0 \\ 0 & 0 & 0 & G_4 & -G_4 \\ 0 & 0 & 0 & -G_4 & G_4 \end{bmatrix} \right\} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \end{bmatrix} + \begin{bmatrix} -G_1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} V_i = 0 \tag{3}$$

where $G_i = 1/R_i$ and $G_T = G_1 + G_2 + G_3$.

It can also be observed that no nodes can be deleted without increasing the degree of the system as there are no zero

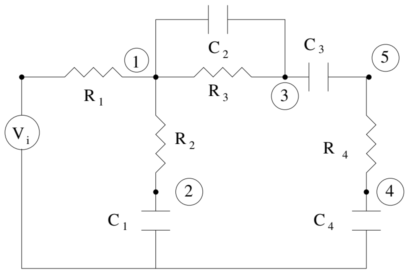

Figure 4. Experimental Circuit

columns in the first order matrix, the coefficient matrix of s in equation 3. Since the degree of the system can be increased up to the second degree, some nodes can still be deleted from the system of equations. Thus, the cost of deleting each node and the equations that can be utilized for this purpose are determined as shown in Table 2. It can be also observed that Node 3 cannot be deleted from the system of equations since all the equations Node 3 appears in involve its differentiation.

After re-arranging the matrix entries, the candidate set of nodes for deletion is $\{1,2,4,5\}$. The largest non-singular submatrix within the candidate set indicates which nodes can be deleted independently from each other. The selected set of nodes to be deleted from the equation is $\{2,5\}$; the other alternative, $\{1,4\}$, yields a symmetrical solution. After expressing the voltages of nodes 2 and 5 in terms of the voltages of nodes 1, 3 and 4, the final system of equations reduces to:

$$\left\{\begin{bmatrix} \frac{C_1 C_2}{G_2} & 0 & 0 \\ 0 & 0 & -\frac{C_3 C_4}{G_4} \\ 0 & 0 & \frac{C_3 C_4}{G_4} \end{bmatrix} s^2 + \begin{bmatrix} \frac{C_1 G_T}{G_2}+C_2 & \frac{C_1 G_3}{G_2} & 0 \\ -C_2 & C_2+C_3 & -C_3 \\ 0 & -C_3 & C_3+C_4 \end{bmatrix} s + \begin{bmatrix} G_1+G_3 & -G_3 & 0 \\ -G_3 & G_3 & 0 \\ 0 & 0 & 0 \end{bmatrix}\right\} \begin{bmatrix} V_1 \\ V_3 \\ V_4 \end{bmatrix} + \left\{\begin{bmatrix} -\frac{C_1 G_1}{G_2} \\ 0 \\ 0 \end{bmatrix} s + \begin{bmatrix} -G_1 \\ 0 \\ 0 \end{bmatrix}\right\} V_i = 0$$

Since the hardware budget for the on-line test circuit is reached at this point, no more nodes can be eliminated. The next step is to select a set of the remaining three equations to implement. As can be seen from Table 3, components C_1, R_1, and R_2 are covered by Row 1 only. Therefore, the corresponding equation has to be implemented. The remaining components, C_3, R_4, and C_4 are covered by rows 2 and 3. Since the equation corresponding to row 2 also is affected by variations in G_3 and C_2, the 3^{rd} row equation

Node	1	2	3	4	5
Deletion Cost	1	1	∞	1	1
Equations	2	1	N/A	5	4

Table 2. The Cost of Deleting Each Node

Equation	Components
1	C_1, C_2, R_1, R_2, R_3
2	C_2, C_3, C_4, R_3, R_4
3	C_3, C_4, R_4

Table 3. Component Coverages of Equations

is selected to provide an even coverage in terms of all the components. Weights of 1.0 and 0.6 are assigned to equations 1 and 3 respectively in order to even out the sensitivity of the invariant with respect to component variations. The final detection circuit can be seen in Figure 1.

The threshold for alarming a failure and the minimum detectable deviation for each component are determined using both Monte-Carlo and mathematical analyses. In the mathematical tolerance analysis, OPAMPs are assumed to be ideal since mathematical models are needed to conduct this analysis, whereas in circuit simulations, OPAMPs are replaced by their transistor level circuits in order to include the effects of process variations on the OPAMP parameters. The distributions of the components are assumed to have a mean equal to their nominal value and a standard deviation equal to 1/3 of their tolerance of 6%. Both analyses yield very similar results for detectable component variations even though the mathematical analysis cannot include the effects of process variations on OPAMP transistor-level parameters. This result is to be expected, since OPAMPs, when used in feedback-based configurations, are highly robust to process variations.

Figure 5 shows the upper 3σ point of the frequency domain response of the fault-free invariant and the threshold determined from this response. The threshold is set at the 4σ point in order to reduce the probability of a false alarm during the production testing phase. The lower 3σ response of the invariant with a 20% deviation in R_1 is also shown in Figure 5. The range of frequencies that this response exceeds the threshold constitutes the detection window for this

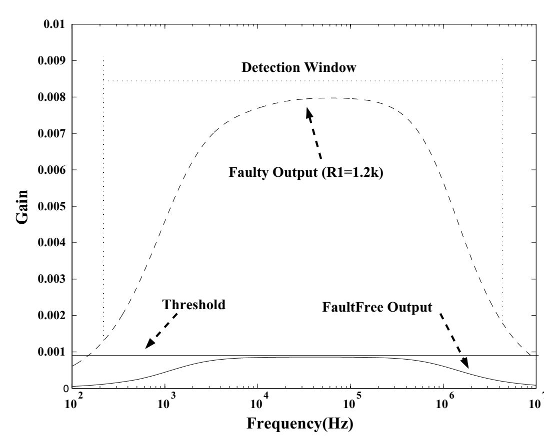

Figure 5. Frequency Domain Response of the Invariant

Proceedings of the 2002 IEEE International Conference on Computer Design: VLSI in Computers and Processors (ICCD'02)
1063-6404/02 $17.00 © 2002 **IEEE**

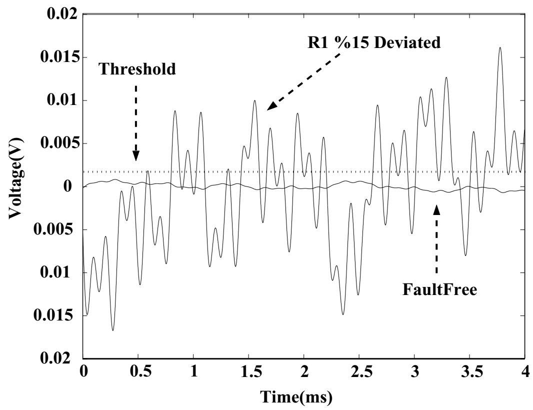

Figure 6. Invariant Response to a Multi-Tone Signal

fault. Whereas yield loss due to a false alarm is considerably low (0.1%) due to a higher threshold, even a 20% deviation in resistor R_1 is observable. If desired, yield loss can further be reduced by allowing an increase in the minimum detection value of the components without reducing the detection capability since the goal is to alarm catastrophic system failures.

Figure 6 shows the upper 3σ point of the fault free response and the lower 3σ point of the response with a 15% deviation in R_1 to a multi-tone waveform. The worst case deviation that causes the invariant response to exceed the threshold for each component is computed through the proposed statistical analysis. Table 4 shows the results of both Monte-Carlo and mathematical analyses.

Table 5 provides a comparison of the proposed method with the previous approaches in concurrent test of analog circuits [2, 3, 5] in terms of fault coverage (FC), hardware overhead (HO) and performance overhead (PO). Since scanty detailed data for fault coverage is provided in [3, 5], a fault coverage comparison can be made only in terms of the percentage of the components, the deviation in which result in a failure alarm. The hardware overhead is compared in terms of the degree of the detection circuit and the performance overhead is compared in terms of the percentage of the nodes that need to be observed.

No experimental data is provided in [2, 3] to compare fault coverage. Hardware overhead and performance overhead of both methods presented in [2, 3] and [5] cannot be controlled; the method presented in [2, 3] requires observation of all state variables some of which may not be physically accessible. In contrast to the previous approaches, the proposed method guarantees full coverage all all the circuit components while minimizing the performance overhead and keeping the hardware overhead within designer-imposed limits.

Comp	R_1	R_2	R_3	R_4	C_1	C_2	C_3	C_4
MC	0.12	0.12	0.16	0.16	0.18	0.20	0.20	0.40
MA	0.11	0.12	0.15	0.15	0.18	0.22	0.22	0.38

Table 4. Minimum Detectable Component Deviations

	[2, 3]	[5]	Proposed
FC	N/A	67%	100%
HO	1-fixed	1-fixed	2-adjustable
PO	100%(+)	100%	60% - adjustable

Table 5. Comparison with Previous Approaches

6 Conclusion

Concurrent testing for analog circuits is an essential element for safety critical systems since any system employs some sort of analog circuitry to interact with the environment. The difficulty of off-line analog test, due to parameter tolerances and continuity of analog signal attributes, is exacerbated by the inability of controlling the input signals in concurrent test. Further, stringent coverage constraints need to be ensured for a viable approach while hardware overhead of the testing circuit and the performance overhead of observing internal nodes need to be kept within acceptable bounds. In addition, in order to keep the rate of costly false alarms due to variations in the circuit components as low as possible, parameter tolerances must be incorporated into the design flow of the detection circuit.

In this paper, we present a methodology for automatically designing a detection hardware for the concurrent test of linear analog circuits. The goal is to keep the hardware overhead within a specified bound while providing 100% fault coverage and optimizing the detection circuit in terms of loading overhead.

The proposed methodology is illustrated on an RC filter circuit. The suggested approach scales effectively as the hardware overhead of the detection circuitry does not increase with increasing circuit size, since it only depends on the number of differentiations in the modified nodal analysis, bounded by a maximum of two for linear analog circuits. Experimental results confirm that an effective detection circuit for concurrent analog test can be designed even while constraining the hardware overhead to a tight budget.

References

[1] I. Bayraktaroglu and A. Orailoglu. Concurrent test for digital linear systems. *IEEE TCAD*, 20(9):1132–1142, September 2001.

[2] A. Chatterjee. Checksum-based concurrent error detection in linear analog circuits with second and higher order stages. In *IEEE VTS*, pages 286–291, 1992.

[3] A. Chatterjee. Concurrent error detection and fault tolerance in linear analog circuits using continuous checksums. *IEEE TVLSI*, 1(2):138–150, June 1993.

[4] M. Etzel and W. Jenkins. Redundant residue number systems for error detection and correction in digital filters. *IEEE TASSP*, 28(5):538–545, October 1980.

[5] E. Simeu, A. Peters, and I. Rayane. Automatic design of optimal concurrent fault detector for linear analog systems. In *FTCS*, pages 184–191, 1999.

[6] G. Taguchi. Performance analysis design. *International Journal of Production Research*, 16:521–530, 1978.

Proceedings of the 2002 IEEE International Conference on Computer Design: VLSI in Computers and Processors (ICCD'02)
1063-6404/02 $17.00 © 2002 **IEEE**