Analog Testability Analysis and Fault Diagnosis using Behavioral Modeling

Edward Liu* William Kao* Eric Felt Alberto Sangiovanni-Vincentelli

Cadence Design Systems, Inc.*
Department of EECS, University of California, Berkeley, CA 94720

Abstract

This paper presents an efficient strategy for testability analysis and fault diagnosis of analog circuits using behavioral models. A key contribution is a new algorithm for determining analog testability. Experimentally, we determined the testability and faults of a fabricated 10 bit digital-to-analog converter modeled using the analog hardware description language, Cadence-AHDL. Also, we applied the testability analysis at the circuit level using SPICE sensitivity analysis.

1 Introduction

The goal for testability and fault diagnosis is to identify fault location and severity from measurements. By identifying the faults, we can identify design weak spots and problems in processing, thus pinpointing the critical areas for improvement in the next design iteration. In this context, the **testability** of a circuit corresponds to the extent faults can be identified from measurements. **Fault diagnosis** corresponds to the algorithm for computing the severity of faults from measurements.

In general, fault diagnosis for analog circuits is difficult because few general fault models can be applied to all analog circuits and few simulators are available for efficient analog fault simulation. Thus, the relationship between output errors and component errors (faults) cannot be easily determined for analog circuits. In the case where the component error is small as in mismatches or small variations in component parameters, a linear error model can be used[1]. This model relates the output errors as a linear function of the component errors,

$$s = Uc, \tag{1}$$

where $s \in R^n$ is the output errors, $c \in R^m$ is the component errors, $U \in R^{nxm}$ is the sensitivity matrix, n is the number of output measurements, and m is the number of components. Column i of U is the sensitivity of the output to component i, $c(i)$.

Using the linear error model, an algorithm for determining testability will be presented in Section 2, followed by a procedure for fault diagnosis in Section 3. In Section 4, behavioral simulation and modeling for testability analysis will be discussed. Finally, experimental results are presented in Section 5, followed by a conclusion in Section 6.

2 Testability analysis

For the linear error model[1], errors can be diagnosed by solving the system of equations in (1) using linear regression[2] when certain conditions are met. Using linear regression, we compute the optimal least square estimate of component errors,

$$\hat{c} = (U^TU)^{-1}U^Ts, \tag{2}$$

provided $(U^TU)^{-1}$ exists. To determine the extent to which errors can be diagnosed, let's consider first the number of observed outputs, n, and the number of components, m. Obviously, the number of components must be smaller than or equal to the number of observed outputs for $(U^TU)^{-1}$ to exist. Otherwise, U^TU is singular and not invertible. Therefore, the condition $n \geq m$ must hold.

Secondly, once U is determined from the behavioral model, we must verify that the sensitivity vectors in U are linearly independent from one another. Otherwise, the dependency will corrupt the inverse of U^TU because of ill-conditioning. To illustrate, suppose the first two sensitivity vectors are linearly dependent for a device under test.

$$\begin{bmatrix} s(1) \\ s(2) \\ s(3) \end{bmatrix} = \begin{bmatrix} 1 & 2 & 0 \\ 1 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c(1) \\ c(2) \\ c(3) \end{bmatrix} \tag{3}$$

where $n = m = 3$. In this case, U^TU is singular, thus, not invertible.

2.1 Ambiguity groups

Components whose sensitivity vectors are dependent are said to fall into the same **ambiguity group**[3, 4, 5]. There can be more than one ambiguity group in a circuit. Each component can either be independent or belong to only one group for the obvious reason that if the component belongs to two groups, the union of the two groups is a group. As a result, the collection of ambiguity groups is unique.

The testability of a circuit decreases with the number of ambiguity groups and the number of components in each group. Thus, given a sensitivity matrix, U, the goal in testability analysis is to find the collection of ambiguity groups. In [5], the groups are found by testing if sensitivity vectors are dependent two vectors at a time, then three vectors at a time, and so on until the CPU time limit is exceeded. Due to the exponential complexity of such brute force approach, its usefulness is limited.

In this paper, we present an efficient approach for determining ambiguity groups. Suppose U has dependent columns, then its null space is non-empty such that,

$$UN = 0 \tag{4}$$

where $N \in R^{mxr}$ is a matrix with r independent column vectors that spans the null space of U. Notice that non-zero entries in N are important because they indicate that the corresponding components are in ambiguity groups. Clearly, a component i belongs to an ambiguity group if and only if row i of N has a non-zero entry. Furthermore, we have the following theorem.

Theorem 1 *Components i and j are in the same ambiguity group iff rows i and j of N are non-zero and not orthogonal to each other.*

Proof. Suppose components i and j are not in the same ambiguity group and not orthogonal. Then, because they are not in the same group, there exists an orthonormal matrix $M \in R^{mxm}$ such that if any entry in $N_i M$ is non-zero, then the corresponding entry in $N_j M$ is zero, or vice-versa, where $N_i \in R^{1xm}$ and $N_j \in R^{1xm}$ are rows i and j of N. It follows that

$$(N_i M)(N_j M)^T = N_i M M^T N_j^T = 0 \tag{5}$$

which implies that

$$N_i N_j^T = 0 \tag{6}$$

since $MM^T = I$. Thus, N_i and N_j are orthogonal, which results in contradiction and completes the proof.

From Theorem 1, it follows that the components fall into the same group if their corresponding row vectors of N are non-zero and not orthogonal. In other words, we have the following corollary,

Corollary 1 *The number of ambiguity groups is equal to the number of orthogonal subspaces spanned by the rows of N.*

The null space of U can be computed using singular value decomposition (SVD) or Gaussian elimination. In the case of SVD, we first compute $U^T U$, followed by SVD,

$$U^T U = X_1 X_2 N^T \tag{7}$$

where X_1 and X_2 are matrices not relevant to the current discussion, and N spans the null space of $U^T U$. Since $U^T U N = 0$, $UN = 0$, so N is the null space of U also. The reason for computing $U^T U$ in (7) is that U often has many more rows than columns, thus, computing N for a smaller matrix $U^T U$ is more efficient.

Because computing N and checking the rows of N for pair-wise orthogonality can be performed in polynomial time, the new approach is faster than the previous approach, particularly when the circuit size increases. In summary, the approach for testability analysis is as follows.

- 1. Given a sensitivity matrix U.
- 2. Find null space of U, N, using singular value decomposition (7) or Gaussian elimination. Let group number $g = 1$.
- 3. Remove the first non-zero row of N and assign to group number g.
- 4. Check if any remaining rows are orthogonal. If not, assign them to group g and remove.
- 5. Increment g and repeat 3 until all rows are removed.

3 Fault diagnosis

Using the results from testability analysis presented in Section 2, we present in this section the entire fault diagnosis procedure as follows:

- 1. Measure n outputs, where $n \geq m$.
- 2. Compute sensitivity matrix U from behavioral model.
- 3. Analyze U for ambiguity groups.
- 4. For each group, set one component as reference and delete the corresponding column from U. Repeat 3 until no ambiguity groups remain.
- 5. Compute error estimates $\hat{c}$ using linear regression given by (2).
- 6. Check residual $s - U\hat{c}$. Estimate $\hat{c}$ is good if residual is small.

Remark: In step 4 of the above procedure, we eliminate from U redundant sensitivity vectors. The components that corresponds to the eliminated vectors are considered reference components that have no errors.

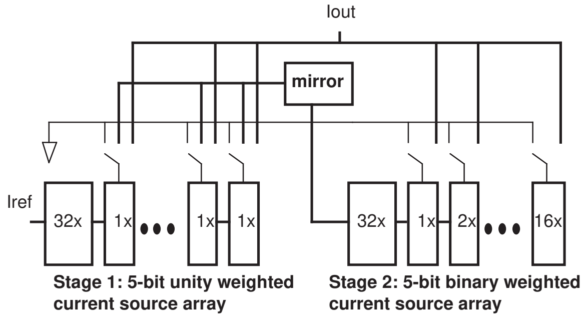

Figure 1: Block diagram of 10 bit DAC

4 Behavioral simulation and modeling

In general, behavioral models capture input-output behavior of analog circuits in a mathematical model without implementation details. They are used to decrease the design time of analog circuits. For instance, in top-down design, designers can explore system behavior quickly without investing time in actual implementation of the components; thus, allowing rapid design space explorations[6]. In bottom-up verifications, designers can verify system implementations quickly because behavioral models are usually simple and quick to evaluate[7]. In fault diagnosis, designers are interested in finding components errors in systems such as current mirror mismatches in DAC or comparator offsets in ADC. Computing the sensitivities in such large circuits using circuit simulators would be prohibitively expensive in time; therefore, the simulation time reduction offered by behavioral modeling makes fault diagnosis feasible in large circuits.

To illustrate, Figure 1 shows a block diagram for a 10 bit interpolative DAC with 38 current mirror components each characterized by its input to output current ratio. The behavior of the converter can be abstracted by the following mathematical equations:

$$I_{o1} = \sum_{i=0}^{MSB-1} k_{1(i)} I_{ref} \tag{8}$$

$$I_{o2} = k_m k_{1(MSB)} I_{ref} \tag{9}$$

$$x = \sum_{i=0}^{4} f(LSB, i) k_{2(2^i)} \tag{10}$$

$$I_{out} = I_{o2} * x + I_{o1} \tag{11}$$

where MSB is the 5 bit integer corresponding to the most significant bits, LSB is the 5 bit integer corresponding to the least significant bits, $f(LSB, i)$ returns unity if the $(i+1)^{th}$ least significant bit in LSB is one, else zero, and k_{1i}, k_{2i}, and k_m corresponds to the input to output current ratio for each component (see Section 5 for values).

5 Experimental results

Experimentally, we have determined the integral nonlinearity (INL) testability of the 10 bit DAC shown in Figure 1 using 5 seconds of DEC 5000 CPU time. There are two ambiguity groups, where one group consists of components **k100** to **k131**, and the other group consists of components **k201**, **k202**, **k204**, **k208**, **k216**, and **km**. Components **k100** to **k131** correspond to the input to output current ratio of the set of 32 unity weighted current mirrors in the first stage. They have value $\frac{1}{32} = 0.03125$, plus a small number representing mismatches. Components **k201**, **k202**, **k204**, **k208**, **k216** correspond to the input to output current ratios of the set of 5 binary weighted current mirrors in the second stage. They have values 0.03125, 0.0625, 0.125, 0.25, and 0.5, respectively, plus mismatches. Component **km** corresponds to the input to output current ratio of the mirror connecting the two stages. It has value 1.0, plus mismatch.

By eliminating **k100** in the first and **k201** in the second group from the sensitivity matrix, we eliminated all ambiguity groups. Then, we diagnosed the component errors for a fabricated converter using linear regression given in (2). Component errors extracted (Figure 3) from measured INL (Figure 2) is fitted into the behavioral model (Appendix). The INL predicted from the behavioral model is plotted along with the measured data (Figure 2), while the difference (residual) is plotted in Figure 4. With maximum at 0.15 LSB, the small amount of residual indicates that the behavioral model can capture mismatch effects effectively. Furthermore, it is seen from Figure 3 that most of the INL are contributed by two faulty components; namely, the second largest binary current source **k208** and the mirror connecting the two stages **km**.

The testability algorithm can also be used directly with SPICE sensitivity analysis to determine the ambiguity groups in a circuit if the circuit is sufficiently small. As an example, consider the 7-RC ladder circuit shown in Figure 5. The circuit is tested by applying a sequence of frequencies at V_{in} and observing V_{out} at each frequency. The ambiguity groups for this circuit are identified in Table 2. The total CPU time for measuring the sensitivities at 91 frequencies and determining the ambiguity groups is less than 1.0 second on at DECstation 5000/260 running Ultrix.

The ambiguity groups for this circuit can be understood intuitively as follows. Consider a simpler 2-RC ladder, which has a two-pole response. The error signatures of R_1 and C_2 are shown in Figure 6. These error signatures are linearly dependent for all frequencies, since both components change the poles of the filter in the same way. Thus R_1 and C_2 are in the same ambiguity group.

6 Conclusion

We have presented an efficient strategy for testability analysis and fault diagnosis of analog circuits using behavioral models. A key contribution is a new algorithm for determining analog testability. Experimentally, we determined the testability and faults of a fabricated 10 bit digital-to-analog converter modeled using the analog hardware description language, Cadence-AHDL. Also, we applied the testability analysis at the circuit level using SPICE sensitivity analysis.

7 Appendix

Cadence-AHDL is an analog hardware description language for behavioral modeling and simulation. The circuit behavior is expressed using the available language constructs including, but not limited to, **assignments**, **binary operators**, and **if-then-else**. Data types includes, but are not limited to, **real** and **integer**. To

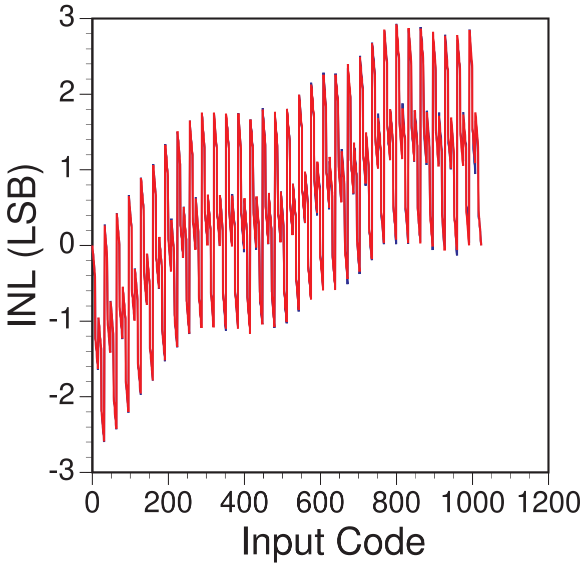

Figure 2: Measured and predicted INL from 10 bit DAC

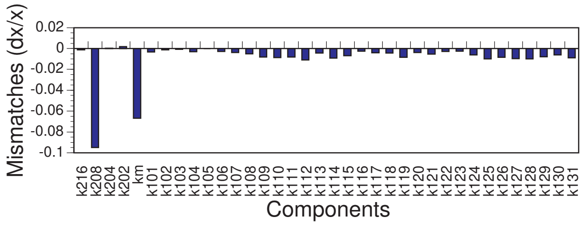

Figure 3: Component errors of 10 bit DAC

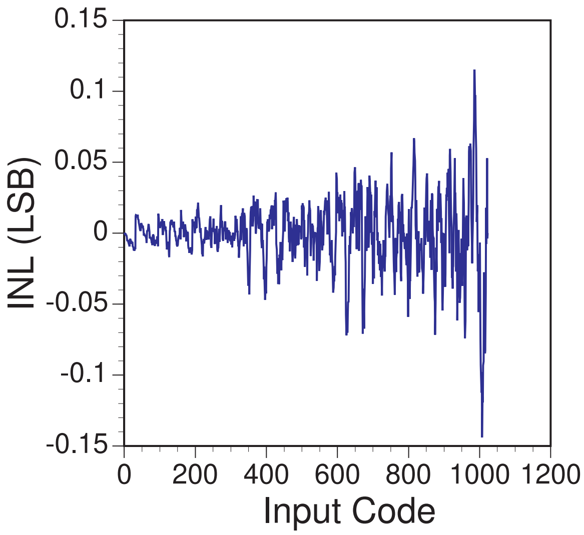

Figure 4: Residual is difference between measured and predicted

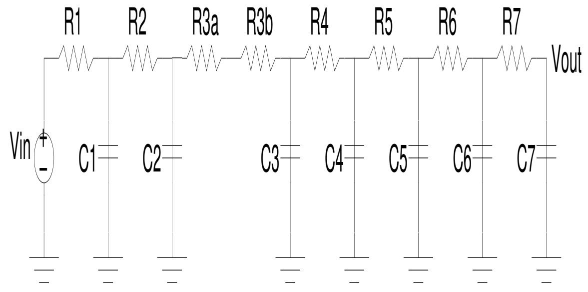

Figure 5: Schematic of SPICE Example

Component	Size
R_1,R_2,R_4,R_5,R_6,R_7	200 Ω
R_{3a},R_{3b}	100 Ω
C_1,C_2,C_3,C_4,C_5,C_6,C_7	10 pF

Table 1: Component Sizes for SPICE Example

Group	Components
1	R_1,C_7,vin_{AC}
2	R_2,C_6
3	R_{3a},R_{3b},C_5
4	R_4,C_4
5	R_5,C_3
6	R_6,C_2
7	R_7,C_1
none	$vinDC$

Table 2: Ambiguity Groups for SPICE Example

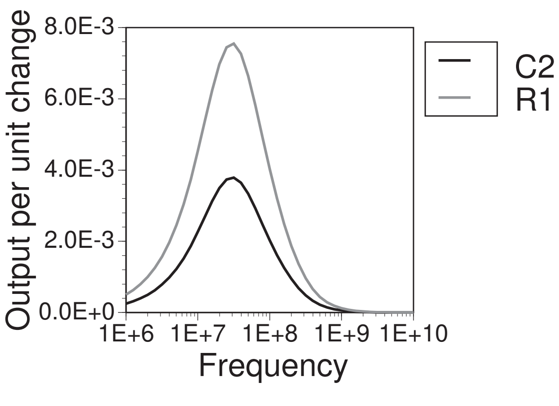

Figure 6: Comparison of signatures of R1 and C2

illustrate the features of this hardware description language, we provide below a partial listing of a model for a fabricated 10 bit two stage interpolative DAC (Figure 1), with five bits in the first stage and five bits in the second stage. The key components in this DAC are the current mirrors, which are modeled by input to output current ratios **k100** to **k131**, **k201**, **k202**, **k204**, **k208**, **k216**, and **km**. Using this model, we compute the nominal converter outputs for all input codes using DC analysis in the **Spectre** simulator. In addition, we compute the sensitivity to the ratios by perturbing the ratios, simulating the output, and calculating the sensitivities using finite difference.

```
//
// D-A Converter
//   10 bit D/A converter modeled after Chip55_8v
//
module da(inputCode, converterOutput, currentReference)
node [V,I] inputCode, converterOutput, currentReference;
{
  real v1, G1, R1; int c, lsb, msb;
  real io1, io2, vo3, x, vcode;
  real k100, k101, k102, k103, k104, k105,
    ...
    ...
    k130, k131;
  real k201, k202, k204, k208, k216, km, xn;
  init {
    R1 = 1k; G1 = 1/R1; x = 1.0/32.0;
    k100 = x; k101 = x; k102 = x;
    ...
    ...
    k130 = x; k131 = x;
    k216 = 1.0/2; k208 = 1.0/4;
    k204 = 1.0/8; k202 = 1.0/16;
    k201 = 1.0/32; km = 1.0;
  }
  every {
    vcode = V(inputCode) * 1024;
    v1 = I(currentReference) * R1;
    V(currentReference) <- v1;
    c = vcode;
    msb = c / 32;
    lsb = c - msb * 32;
    x = 0;
    if (msb > 0) { x += k100; xn = k101; }
    ....
    ....
    if (msb >30) { x += k130; xn = k131; }
    io1 = x * G1 * v1;
    io2 = xn * km * G1 * v1;
    x = 0;
    if (lsb >=16) { x += k216; lsb -=16; }
    if (lsb >= 8) { x += k208; lsb -= 8; }
    if (lsb >= 4) { x += k204; lsb -= 4; }
    if (lsb >= 2) { x += k202; lsb -= 2; }
    if (lsb >= 1) { x += k201; lsb -= 1; }
    vo3 = -(io2 * x + io1);
    V(converterOutput) <- vo3;
  }
}
```

References

[1] T. Souders and G. Stenbakken. Modeling and test point selection for data converter testing. *IEEE International Test Conference*, 1985.

[2] D. Brillinger. *Time Series Data Analysis and Theory, Expanded Edition*. McGraw-Hill, New York, 1981.

[3] J. Bandler and A. Salama. Fault diagnosis of analog circuits. *Proc. of the IEEE*, vol. 73, August 1985.

[4] N. Sen and R. Saeks. Fault diagnosis for linear system via multi-frequency measurements. *IEEE Trans. on Circuits and Systems*, CAS-26, 1979.

[5] G. Stenbakken, T. Souders, and G. Stewart. Ambiguity groups and testability. *IEEE Trans. on Instrumentation and Measurement*, vol. 38, October 1989.

[6] H. Chang, A. Sangiovanni-Vincentelli, F. Balarin, E. Charbon, U. Choudhury, G. Jusuf, E. Liu, E. Malavasi, R. Neff, and P. Gray. A top-down, constraint-driven design methodology for analog integrated circuits. In *Proc. IEEE Custom Integrated Circuits Conference*, pages 841–846, May 1992.

[7] E. Liu, H. Chang, and A. Sangiovanni-Vincentelli. Analog system verification in the presence of parasitics using behavioral simulation. In *Proc. Design Automation Conference*, June 1993.