Nyquist Data Converter Testing and Yield Analysis using Behavioral Simulation*

Edward W. Y. Liu Alberto L. Sangiovanni-Vincentelli

Department of EECS, University of California, Berkeley, CA 94720

Abstract

This paper presents a strategy for testing all DC performance of Nyquist data converters including offset error, full scale gain error, integral nonlinearity, and differential nonlinearity. In contrast to previous testing strategies based on linear models that require accurate measurements of circuit performance, our strategy uses a simpler measurement to verify that a circuit performance parameter falls within certain detection thresholds in the presence of measurement noise. Using the proposed strategy, we can evaluate tradeoffs between test set size, test coverage, detection thresholds, measurement noise, chip performance, and estimated yield. Our results support the obvious that smaller measurement noise, stricter detection thresholds, and lower chip performance would require smaller test set and reduce test time. Stricter detection thresholds, on the other hand, would decrease estimated yield.

1 Introduction

Data converters are commodity products, yet their testing is very expensive. Test engineers use expensive equipment to take accurate measurements against background noise to characterize converters over a variety of operating conditions such as temperature and supply ranges. Furthermore, engineering time needed to develop unique test software for each product adds substantially to the overall testing cost.

A behavioral model that captures the converter behavior can provide critical information for design engineers to evaluate the testability of the design at an early design stage and for test engineers to choose the optimum testing strategy after design. From the information contained in the behavioral model, engineers can evaluate the tradeoffs between *test set size*, *test coverage*, *detection thresholds*, *measurement noise*, *chip performance*, and *estimated yield*. To achieve this goal, we propose a new converter testing strategy for data converters from a behavioral model[1]. We present previous work in Section 2, a new testing strategy in Section 3, a yield analysis algorithm in Section 4, and experimental results in Section 5.

2 Previous Work

In [2, 3] a *linear model* for data converters along with a test selection strategy was presented. The model for an N bit A/D converter represents the $2^N - 1$ transition points of the converter as a linear function of the component errors,

$$t = S_v v \tag{1}$$

where t is an $2^N - 1$-dimensional vector that represents the transition points, v is an m-dimensional vector that represents the component errors, and S_v is the $2^N - 1$ by m sensitivity matrix with full column rank.

Assuming no measurement noise, only m linearly independent test points are required to estimate the m model coefficients. The objective is to find the optimal subset of the full set of $2^N - 1$ test points that minimizes the prediction variance of m. This problem falls in the category of **Optimal Design of Experiments**. It has been shown under the **D-Optimality criterion**[4] that in the limit where the number of test points is large, an optimal selection is S_v', an m by m matrix formed by selected rows of S_v. The m model coefficients, v, are then computed as

$$v = [S_v']^{-1} t' \tag{2}$$

where t' are the selected measurements. Under this criterion, the prediction variance of v is minimized by maximizing $|S_v'^T S_v'|$ where $|\cdot|$ is the determinant.

While the D-Optimality criterion gives the optimal solution, the computational complexity prevents its use for large models. Stenbakken[5] introduced an algorithm for a near optimal solution based on QR factorization with pivoting. The algorithm first chooses the row of S_v with the largest norm, orthogonalizing all remaining rows to it using a modified Gram-Schmidt orthogonalization procedure, then choosing the row of largest norm of those remaining, and repeating the process. The algorithm records the row indices during pivoting, and chooses the m pivot rows of the initial S_v for S_v'.

The main drawback of the previous testing strategy for data converters lies in the difficulty of making accurate measurements. For example, an A/D produces a discrete output code from a continuous input. Under the previous testing strategy, we measure the transitions between adjacent output codes using a bisection search algorithm[6]. In this algorithm, we guess initial lower bound, $t_l(i)$, and upper bound, $t_u(i)$, for the transition $t(i)$. If the outputs of the converter for $t_l(i)$ and $t_u(i)$ are $o_1 \leq i$ and $o_2 \geq i+1$, respectively, then $t(i)$ must lie within the bounds. By moving the bounds closer iteratively until convergence, we can determine $t(i)$ up to an accuracy limited by the measurement noise. Because the algorithm has complexity of $O(log_2(\frac{1}{\alpha}))$, where α is the required accuracy in fraction of the smallest step size (LSB), we need many iterations requiring long test time for accurate measurements.

Moreover, measurement noise poses a significant problem because it corrupts the estimated parameters. To reduce measurement noise problems, Hemink[7] presented algorithms for testability analysis and optimal test selection in the presence of measurement noise. On the other hand, Souders[8] proposed adding more test points to reduce noise effects by creating an overdetermined system of equations to be solved by least square techniques. Although additional test points reduce noise effects, no algorithm has been presented so far for determining the number of extra tests needed or selecting the additional tests based on the amount of measurement noise.

*This project is supported by the Semiconductor Research Corporation under grant 92-DC-008. Its support is gratefully acknowledged.

Another drawback of the previous approach is the large amount of computation required for solving the system of equations for each chip under worst case operating conditions such as temperature and supply extremes. Also, the lack of consideration for tradeoffs between test set size, chip performance, test coverage, and yield reduces the usefulness of the previous strategies.

3 New Testing Strategy

Data converters are binned with respect to their performance and sold at prices accordingly; therefore, we need a strategy to tradeoff tested chip performance and test costs, as well as other parameters such as test coverage, measurement noise, and estimated yield. Decreasing test coverage, smaller measurement noise, stricter detection thresholds, and lower chip performance would require smaller test set and less test time. Stricter detection thresholds, on the other hand, would decrease estimated yield. An optimal choice of these parameters will lead to the most cost effective testing solution. In this section, we present a testing strategy that allows tradeoffs between these parameters.

Our strategy focuses on testing all DC performance of Nyquist data converters including offset error, full scale gain error, integral nonlinearity, and differential nonlinearity. To achieve this goal, we present in the next section a behavioral model[1] for data converters that is useful for testing DC performance.

3.1 Behavioral model for data converters

3.1.1 Model derivation

A memoryless Nyquist A/D operating at a certain environment such as fixed sampling frequency and temperature can be described by its transfer curve, which plots the output code on the range against the input continuous value on the abscissa. Assuming an A/D has N bits of resolution, is designed to be monotonic and has no missing codes, the transfer curve can be characterized by $2^N - 1$ real numbers in a vector, t, representing the values of the transition points $t(i), i = 1 \dots 2^N - 1$. Due to statistical process variations, the transfer curve t has a statistical distribution. Due to noise, an A/D produces a wrong output with a non-zero probability. For example, if the input is near a transition point, the probability of getting a wrong output code, the code at the other side of the transition point, is non-zero. As a result, the noise effects depend on the input, and they become more significant as the input gets closer to a transition point. Mathematically, this is represented by a probability distribution function of the output codes for any input value. Our behavioral model captures this noise effect by a joint probability density function of the input V_{in} and the output code for a given realization of the transfer curve $\hat{t}$,

$$f(code, V_{in}, \hat{t}) \tag{3}$$

From the above equation, which describes the behavior of any memoryless A/D due to noise and process variations completely, we can extract all information about the converter.

In our model, noise and process variation effects are separated. The distribution t captures process variation effects, while the joint density function $f(\cdot)$ captures noise effects. The function $f(\cdot)$ is computed using either direct techniques[9] or Monte Carlo techniques. The process variation effects can be derived by making assumption:

Assumption 3.1 *Process variations are the cumulative effects of many integrated circuit fabrication steps. Thus, according to the Central Limit Theorem in probability theory, the distribution of the process variations, v, is approximately multivariate normal.*

Next, we propose a Gaussian model, as well as a non-Gaussian model for the distribution t. The Gaussian model is derived from making the following assumption:

Assumption 3.2 *In general, an A/D is designed to be insensitive to process variations which are relatively small. Thus, the transition points, t, change relatively little with respect to process variations. Hence, we assume that t changes linearly with respect to the process variations:*

$$t = \frac{\partial t}{\partial v} v + \mu_t \tag{4}$$

where v is a random vector of parameters, corresponding to process variations such as offsets and mismatches in components, and $\frac{\partial t}{\partial v}$ is a sensitivity matrix.

From assumption 3.1 and (4), the covariance matrix of t is given by

$$\Sigma_t = \frac{\partial t}{\partial v} \Sigma_v \frac{\partial t}{\partial v}^T \tag{5}$$

where Σ_v is the covariance matrix of v. Thus, the distribution of t is

$$t \sim normal(\mu_t, \Sigma_t) \tag{6}$$

In general, the rank r_t of Σ_t is much less than full rank since not all the transition points are independent. In other words, any realization of t must fall into a space spanned by r_t independent basis vectors. A set of such independent basis vectors can be found by the singular value decomposition of Σ_t. Mathematically,

$$\Sigma_t = U_t \Sigma_{ct} U_t^T \tag{7}$$

where U_t is an $2^N - 1$ x r_t transformation matrix, Σ_{ct} is an r_t x r_t diagonal matrix, and r_t is the rank of Σ_{ct}. The columns of U_t are called the **error signatures** [3] and form a set of orthonormal vectors spanning the **error space**. For storage and computational efficiency t is expressed as the following equation instead of (4),

$$t = U_t c_t + \mu_t \tag{8}$$

where $c_t \sim normal(0, \Sigma_{ct})$ is a small, zero mean multivariate normal distribution with r statistically *independent* variates. The parameters U_t and μ_t are unique for each A/D architecture, while the distribution c represents a mixture of the relevant process variations.

Summarizing, our entire Gaussian A/D model consists of three equations,

$$f(code, V_{in}, \hat{t}), \hat{t} = realization(t) \tag{9}$$

$$t \sim normal(\mu_t, \Sigma_{ct}) = U_t c_t + \mu_t \tag{10}$$

$$c_t \sim normal(0, \Sigma_{ct}) \tag{11}$$

After presenting the complete A/D model, we develop a similar D/A model using the same strategy. The resulting Gaussian D/A model is,

$$t = U_t c_t + \mu_t + z \tag{12}$$

$$c_t \sim normal(0, \Sigma_{ct}) \tag{13}$$

$$z \sim normal(0, \Sigma_z) \tag{14}$$

where t is an n-dimensional vector and z is an n-dimensional vector representing additive noise in D/A outputs.

3.1.2 Calculation of system performance

Using the converter models, we propose a strategy to calculate system performance. The performance of a converter is computed in two steps. First, the converter model parameters are extracted from the circuit. Then, the converter performance is computed using only the model parameters since the model completely capture the converter behavior. The advantage of this strategy is that the model parameters are only extracted once, but used many times to calculate system performance. The following definitions illustrate how system performance parameters can be computed from the proposed D/A model parameters.

Definition 3.1 *The ideal output vector, L, of a D/A converter is an n-dimensional vector, where the i^{th} component of L, $L(i)$, represents the desired output for input code i, $n = 2^N$, and N is the number of bits.*

Definition 3.2 *The output vector, t, of a D/A converter is an n-dimensional vector, where the i^{th} component of t, $t(i)$, is the output for input code i.*

Theoretically, t is sufficient to describe the converter behavior due to process variations. Nevertheless, designers traditionally used a different set of specifications which can be derived easily from t. All of them are described as follows,

Definition 3.3 *Offset error is $t(1) - L(1)$.*

Definition 3.4 *Full scale gain error is $t(n) - L(n) + L(1) - t(1)$.*

The integral nonlinearity is the deviation of the output from the ideal value *after* compensation for the gain and offset errors.

Definition 3.5 *The integral nonlinearity vector, s, of a D/A converter is an n-dimensional vector*

$$s(i) = at(i) + b - L(i) \tag{15}$$

where a, b are constants such that $s(1) = s(n) = 0$.

Intuitively, Definition 3.5 means that we transform t using constants a and b such that the two end-points are ideal. In this case, we have adopted the end-point method for linear error compensation[10]. The motivation behind the transformation is to compensate for the offset and gain errors before calculation of nonlinearity. In many applications, offset and gain errors are irrelevant as they do not contribute to distortion.

Lemma 3.1 *Integral nonlinearity is given by*

$$s(i) = \left[\frac{L(n) - L(1)}{t(n) - t(1)}\right](t(i) - t(1)) + L(1) - L(i) \tag{16}$$

The sensitivity of the integral nonlinearity to output vector is given by

$$\frac{\partial s}{\partial t}(i,j) = \begin{cases} \frac{(L(n)-L(1))(t(i)-t(n))}{(t(n)-t(1))^2} & i \neq 1, n \quad j = 1 \\ 0 & i \neq 1, n \quad j \neq 1, i, n \\ \frac{L(n)-L(1)}{t(n)-t(1)} & i \neq 1, n \quad j = i \\ 0 & i = 1, n \quad j = 1 \ldots n \\ \frac{(L(n)-L(1))(t(1)-t(i))}{(t(n)-t(1))^2} & i \neq 1, n \quad j = n \end{cases} \tag{17}$$

Proof. Solving for a, b using constraints in Definition 3.5 yield (16). Equation (17) follows from differentiation of (16). ■

Using a first order Taylor approximation of s as a function of t,

$$s \approx \frac{\partial s}{\partial t}(t - \mu_t) + \mu_s = \frac{\partial s}{\partial t} U_t c_t + \mu_s \tag{18}$$

we find the integral nonlinearity has distribution,

$$s \sim normal(\mu_s, \Sigma_s) \tag{19}$$

where $\mu_s = s(t)$, at $t = \mu_t$ using (16) and $\Sigma_s = \frac{\partial s}{\partial t} \Sigma_t \frac{\partial s}{\partial t}^T$ at $s = \mu_s$. Similar to (6), the rank r_s of Σ_s is much less than full rank since not all the INL errors are independent. Any realization of s must fall into a space spanned by r_s independent basis vectors. A set of such independent basis vectors can be found by the singular value decomposition of Σ_s. Mathematically,

$$\Sigma_s = U_s \Sigma_{cs} U_s^T \tag{20}$$

where U_s is a 2^N x r_s transformation matrix, Σ_{cs} is an r_s x r_s diagonal matrix, and r_s is the rank of Σ_{cs}. The columns of U_s are called the **INL signatures** and form a set of orthonormal vectors spanning the **INL space**. For storage and computational efficiency s is expressed as the following equation,

$$s = U_s c_s + \mu_s \tag{21}$$

where $c_s \sim normal(0, \Sigma_{cs})$ is a small, zero mean multivariate normal distribution with r_s statistically *independent* variates.

Besides INL, designers are also interested in differential nonlinearity, DNL, defined as follows,

Definition 3.6 *The differential nonlinearity vector, d, of a D/A converter is an $n-1$-dimensional vector obtained by a first order difference of the integral nonlinearity s.*

$$d = D_+ s \tag{22}$$

where D_+ is an $n-1$ by n matrix called the first order difference operator,

$$D_+(i,j) = \begin{cases} -1 & i = j \\ 1 & i = j - 1 \\ 0 & else \end{cases} \tag{23}$$

As DNL is a linear transformation of INL, most results related to INL carry over to DNL immediately. For instance, the differential nonlinearity has distribution,

$$d \sim normal(\mu_d, \Sigma_d) \tag{24}$$

where $\mu_d = D_+ s$, $\Sigma_d = D_+ \Sigma_s D_+^T$. Any realization of d must fall into a space spanned by r_d independent basis vectors which can be found by the singular value decomposition of Σ_d. Mathematically,

$$\Sigma_d = U_d \Sigma_{cd} U_d^T \tag{25}$$

where U_d is an $2^N - 1$ x r_d transformation matrix, Σ_{cd} is an r_d x r_d diagonal matrix, and r_d is the rank of Σ_{cd}. The columns of U_s are called the **DNL signatures** and form a set of orthonormal vectors spanning the **DNL space**. For storage and computational efficiency d is expressed as the following equation,

$$d = U_d c_d + \mu_d \tag{26}$$

where $c_d \sim normal(0, \Sigma_{cd})$ is a small, zero mean multivariate normal distribution with r_d statistically *independent* variates.

The model proposed in [3] is deterministic and hence does not represent statistical effects. We represent electronic noise as well as process variation effects. Moreover, our model has distributions for INL and DNL. We will show that these are essential to developing a testing strategy that tradeoffs between test set size, test coverage, detection thresholds, measurement noise, chip performance, and estimated yield.

3.2 Measurement and Detection Threshold

As mentioned in Section 2, previous testing strategies[3, 7] have difficulty in accurate measurements of circuit performance such as A/D converter transition points in the presence of measurement noise. To circumvent the problem, we propose a simpler measurement that is robust against noise. In contrast to the traditional approach where the exact value of a performance parameter such as a transition point is measured, we verify that *a circuit performance parameter falls within certain detection thresholds in the presence of measurement noise.* For example, the test for A/D converter transition point, $t(i)$, consists of only *two* A/D conversions for input thresholds $t_l(i)$ and $t_u(i)$. A successful test corresponds to outputs of the converter for $t_l(i)$ and $t_u(i)$ being $o_l \leq i$ and $o_u \geq i+1$, respectively. In this case, the transition point plus uncertainty due to noise, $t(i) + \delta$, will lie within the range given by $(t_l(i), t_u(i))$, where δ is the uncertainty due to noise. As a result, the following inequalities hold,

$$t_l(i) \leq t(i) + \delta \leq t_u(i), \tag{27}$$

$$t_l(i) - \delta \leq t(i) \leq t_u(i) - \delta, \tag{28}$$

$$t_l(i) - max(\delta) \leq t(i) \leq t_u(i) - min(\delta). \tag{29}$$

Assuming noise δ falls in the range of $(-\Delta, \Delta)$, $t(i)$ satisfies the following inequality in the presence of measurement noise (Figure 1),

$$t_l(i) - \Delta \leq t(i) \leq t_u(i) + \Delta. \tag{30}$$

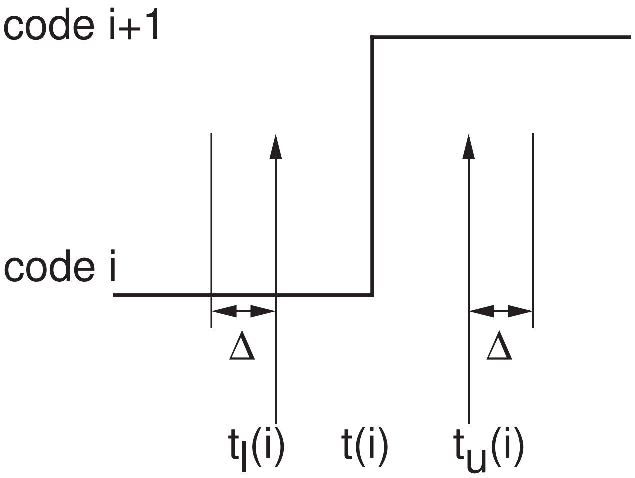

Figure 1: Detection thresholds

3.3 Testing Gain and Offset Errors

Successful tests for gain and offset errors establish bounds on these parameters. Gain and offset errors (Definitions 3.3 and 3.4) are linear functions of the first and last transition points, $t(1)$ and $t(n)$, respectively. We can establish bounds on these errors by choosing tight bounds such as $t_l(1)$, $t_u(1)$, $t_l(n)$, and $t_u(n)$. For example, bounds on offset error, V_{os}, and full scale gain error, V_{ge}, are

$$t_l(1) - \Delta - L(1) \leq V_{os} \leq t_u(i) + \Delta - L(1), \tag{31}$$

$$t_l(n) - L(n) + L(1) - t_u(1) - 2\Delta \leq V_{ge}, \tag{32}$$

$$V_{ge} \leq t_u(n) - L(n) + L(1) - t_l(1) + 2\Delta. \tag{33}$$

3.4 Testing Nonlinearity Errors

Successful tests for integral and differential nonlinearity establish bounds on these parameters for *all* inputs. For a 10 bit converter, there are a total of 1024 possible inputs. Fortunately, it suffices to test a subset, T, of the set of all inputs, S, due to correlations between outputs of data converters. That is, we can establish bounds on all nonlinearity errors by performing tests in T.

We can bound integral and differential nonlinearities for individual inputs using the proposed measurement scheme. The user specifies the **INL test threshold**, s_b, from which we compute the needed test input thresholds using the following equations.

From (16) and (30), we derive inequalities for the integral nonlinearity, $s(i)$,

$$s_l(i) \leq s(i) \leq s_u(i), \tag{34}$$

where

$$s_l(i) = \left[\frac{L(n) - L(1)}{t_u(n) - t_l(1) + 2\Delta}\right](t_l(i) - t_u(1) - 2\Delta) + L(1) - L(i), \tag{35}$$

$$s_u(i) = \left[\frac{L(n) - L(1)}{t_l(n) - t_u(1) - 2\Delta}\right](t_u(i) - t_l(1) + 2\Delta) + L(1) - L(i), \tag{36}$$

$s_l(i) = -s_b$, $s_u(i) = s_b$, and s_b is the INL threshold in the presence of noise. We use (35) and (36) to compute the test inputs $t_l(i)$ and $t_u(i)$ where $i \in 2 \ldots 2^N - 2$. As a rule, the threshold, s_b, should be set as follows,

$$s_b = s'_b + 2\Delta, \tag{37}$$

where s'_b is the INL threshold for ideal noiseless measurements, due to the following approximations,

$$s_l(i) \approx (t_l(i) - t_u(1) - 2\Delta) + L(1) - L(i) \approx -s'_b - 2\Delta \tag{38}$$

$$s_u(i) \approx (t_u(i) - t_l(1) + 2\Delta) + L(1) - L(i) \approx s'_b + 2\Delta \tag{39}$$

The approximations show a linear dependence of the threshold, s_b, on noise and help designers set the INL threshold based on noise estimates. Notice that (38) and (39) are only used as a rule for designers. In testing, we use the exact inequalities (35) and (36) to compute the test inputs $t_l(i)$ and $t_u(i)$ where $i \in 2 \ldots 2^N - 2$. Similar exact inequalities (35) and (36), and approximate inequalities (38) and (39) can be derived for differential nonlinearity.

3.5 Test Selection for Nonlinearity Errors

Each successful test i establishes upper and lower bounds on $s(i)$. Due to correlations between the transition points (Section 3.1), INL at other points may also be bounded. In test selection, we choose the best subset, T, of test points, S, such that all untested INL's are bounded by a **maximum INL performance**, s_m. Let U be the set of unbounded INL. **Test coverage** is defined as

$$1 - \frac{|U|}{|S|} \tag{40}$$

where $|\cdot|$ denotes the cardinality of a set. **Full coverage** refers to the case where U is empty and hence test coverage is one.

We formulate the test selection problem as follows. Let Δ be the magnitude of the measurement noise in (30), s_m be the specified maximum INL performance, s'_b be the INL threshold in the absence of noise, $s_b = s'_b + 2\Delta$ be the INL threshold in the presence of noise, T be the test set, choose the minimum T such that

$$-s_m \leq -s_b \leq s(i) \leq s_b \leq s_m, i \in T \tag{41}$$

$$-s_m \leq s(i) \leq s_m, i \notin T \tag{42}$$

where s is given by a linear transformation of some independent variables in (21),

$$s = U_s c_s + \mu_s \tag{43}$$

where $c_s \sim normal(0, \Sigma_{cs})$ is a zero mean multivariate normal distribution with r_s statistically *independent* variates.

Due to linear dependence, we use **linear programming**[6] to check the bounds on untested INL, $s(i)$, in (42). We formulate the problem as follows. We check that

$$\max_{c_s} s(i) \leq s_m, i \notin T \tag{44}$$

$$\min_{c_s} s(i) \geq -s_m, i \notin T \tag{45}$$

such that

$$-s_b \leq s(j) \leq s_b, j \in T \tag{46}$$

$$-k\sqrt{\Sigma_{cs}(i,i)} \leq c_s(i) \leq k\sqrt{\Sigma_{cs}(i,i)}, i = 1 \ldots r_s, \tag{47}$$

where k is a user specified parameter to limit the distribution, c_s. The limit prevents signatures with small magnitudes from being scaled unrealistically with unbounded c_s in the linear program. Typically, we choose $k = 5$ for a 10σ statistical limit. If c_s is unbounded, then the statistical information provided by the distribution of c_s is lost.

In a different context of delay fault testing[11], a similar problem formulation has been applied to check bounds on delays of digital circuits. In both formulations, the problems seem to be analytically intractable. For example, the simpler problem of checking whether $|T|$ tests provide sufficient bounds on s would require an exponential complexity of

$$O(|S - T| 2^b). \tag{48}$$

where the exponential dependence is due to the linear programming step and $b = \begin{pmatrix} |S| \\ |T| \end{pmatrix}$, where $\begin{pmatrix} x \\ y \end{pmatrix}$ is the number of ways y objects can be chosen from x different objects.

As a result, the optimum test selection problem is likely intractable. Thus, we propose a heuristic solution. First, we rank the tests according to the algorithm proposed in [5] based on QR factorization with pivoting.

- (a) Choose the row of U_s with the largest norm,
- (b) Orthogonalize all remaining rows to it using a modified Gram-Schmidt orthogonalization procedure,
- (c) Choose the row of largest L_2 norm of those remaining,
- (d) Repeat (b) and (c) until all rows are chosen,
- (e) Record the row indices of all chosen rows, and list them in the order they were chosen.

As a result, we produce a list of INL's with decreasing order of importance where the j^{th} item on the list is $s(i)$.

Next, we use the following algorithm to find T.

- (a) Let $j = 1$, and T be empty.
- (b) If j is larger than the number of test points, done.
- (c) Pick the j^{th} item on the list, $s(i)$. Check $s(i)$ for bounds specified in (42) using linear programming.
- (d) If the linear programming is infeasible, quit due to performance specifications too tight.
- (e) If the bounds are not satisfied, add i to T.
- (f) $j = j + 1$ and go to (b).

Unless the specified performance is infeasible, the algorithm guarantees full coverage since every INL is either bounded or constrained by a test. To tradeoff test size against coverage, the algorithm should stop after a certain coverage is achieved.

4 Yield Analysis

Yield, Y, is defined as the ratio of the number of chips working correctly to the total number of chips. Using our behavioral model, we compute Y by integrating the distribution of performance specifications under user specified bounds. For example, if we use only the INL specifications we have

$$Y = \int_{R_1} f(s) ds \tag{49}$$

where $f(s)$ is the probability density function of s and

$$R_1 : -s_m \leq s(i) \leq s_m, i \in S \tag{50}$$

As a lower bound on Y, the **estimated yield**, $\hat{Y}$, is the ratio of the number of chips passing all tests under our proposed strategy to the total number of chips. Notice that the difference between Y and $\hat{Y}$, Y - $\hat{Y}$, corresponds to the fraction of working chips that did not pass the detection thresholds and labeled failed. To minimize this waste of working chips, we would like to minimize the difference while using a strict enough threshold for a small test set.

Assuming the chip passes gain and offset error tests, estimated yield due to INL tests is given by

$$\hat{Y} = \int_{R_2} f(s) ds \tag{51}$$

where $f(s)$ is the probability density function of s and

$$R_2 : -s'_b \leq s(i) \leq s'_b, i \in T \tag{52}$$

Equation (51) supports the obvious that stricter test thresholds would decrease estimated yield, thus increasing the fraction of working chips thrown away. On the other hand, it will be shown (Section 5) that strict test thresholds would decrease test set size. Thus, yield analysis for a particular test set and thresholds provides critical information for engineers to improve design and to choose the optimal testing strategy.

We estimate yield using Monte Carlo integration according to the following algorithm.

- (a) $i = j = 0$, and k is the number of samples.
- (b) Generate a realization of c_s, $\hat{c}_s$, from the distribution of c_s using a random number generator.
- (c) Compute a realization of s, $\hat{s}$, using (21) and $\hat{c}_s$.
- (d) Check if $\hat{s}$ is in R using (52). If so, $j = j + 1$.
- (e) $i = i + 1$. If $i < k$, go to (b), else done.

Estimated yield, $\hat{Y}$, is then computed as $\hat{Y} = \frac{j}{k}$. Because Monte Carlo integration is used, our estimate $\hat{Y}$ improves with the number of trials, k. The variance of our estimate is inversely proportional to k^2.

5 Experimental Results

We have applied the new strategy to the automatic test generation for a 10 bit current-switched, interpolative D/A[12] shown in Figure 2. We used a bottom-up verification strategy[13] to compute the D/A behavioral model parameters from SPICE netlist and expected process variation parameters. Behavioral simulation of the D/A and automatic test generation have been integrated into a single software package to facilitate **design-for-testability**.

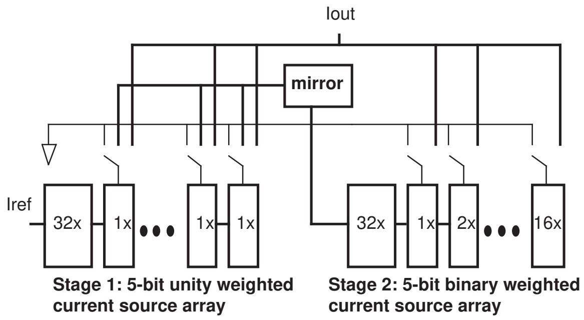

Figure 2: 10 bit current-switched, interpolative D/A architecture

Using the proposed testing strategy, we first determined the tradeoff between maximum INL performance, s_m, and test set size (Figure 3) for $s'_b = 1.5$ LSB, $\Delta = 0.1$ LSB, and full coverage. Notice that test set size varies inversely with INL as expected. Also, the size drops below the rank of 38 for INL larger than 3.2 LSB, where the **rank** is the number of INL signatures. This occurs because some INL signatures with small magnitudes do not contribute significantly to large INL errors. In contrast, previous testing strategy[3] uses the same number of tests as the rank because such strategy does not take into consideration the magnitudes of the signatures. In Figure 4, we plotted the DEC ALPHA CPU times for test generation against INL. Notice that test generation times drops significantly for lower performance chips.

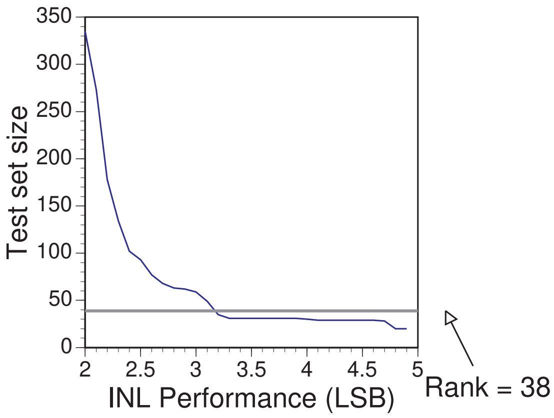

Figure 3: Tradeoff between test set and INL performance

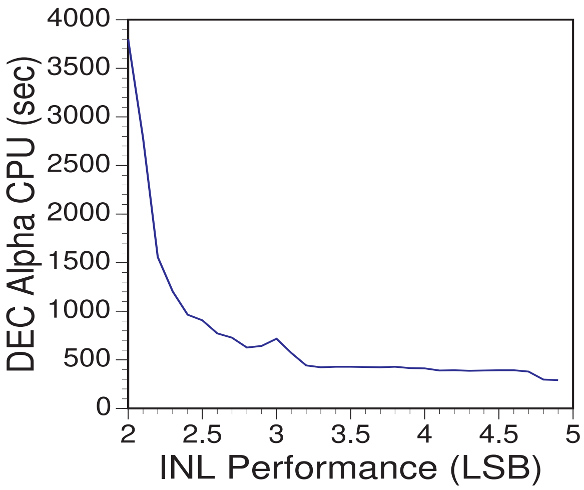

Figure 4: Tradeoff between test generation time and INL performance

Next, we determined the tradeoff between measurement noise and test set size (Figure 5) for $s'_b = 1.5$ LSB, $s_m = 3.0$ LSB, and full coverage. Notice that the size increases somewhat linearly with noise. The average DEC ALPHA CPU times for each data point is 600 seconds.

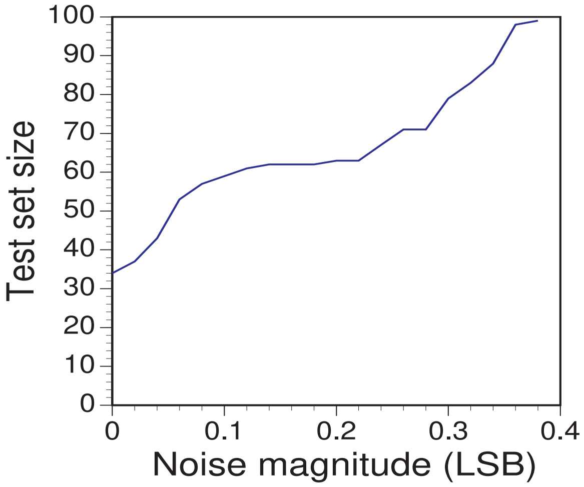

Figure 5: Tradeoff between test set and measurement noise

Also, we determined the tradeoff between INL threshold in the absence of noise, s'_b, test set size, and estimated yield for $\Delta = 0.1$ LSB, $s_m = 3.0$ LSB, and full coverage. Figure 6 plots test set size against s'_b, while Figure 7 plots estimated yield against s'_b. Together, they showed that stricter thresholds decrease estimated yield and test size, yet do not reduce the test size under 31. The reason is that for this level of INL performance, 31 signatures contribute to the INL, while the remaining 7 signatures are insignificant. Thus, 31 or more independent tests are needed to guarantee INL performance regardless of INL thresholds. As a result, we should choose the highest threshold, $s_b = 1.3$ LSB, such that test size is 31 for highest estimated yield and better noise tolerance. The average DEC ALPHA CPU times for each data point is 582 seconds.

Next, we apply the proposed testing strategy to the actual testing of twelve 10-bit D/A converter chips of the same design synthesized by a top-down design methodology[12]. Figure 8 shows the measured INL for all 1024 input codes for two sample silicon chips. For an INL performance specification of 3 LSB, an INL detection threshold of 2 LSB, an assumed measurement noise of 0.1 LSB, and expected process variation parameters, the number of needed tests was computed to be 76 using a total of 446 seconds of DEC 5000 CPU seconds. The first 5 test inputs, in descending order of importance, are 479, 736, 352, 255, and 928. The first failed chip was detected at test number 1 using input code 479. Another was detected at test number 37 using input code 762. The last two failed chips were detected at test number 2 using input code 736. The INL for all input codes for the last failed chip is shown in Figure 8 which shows clearly that the INL exceeds 2 LSB for input code 736.

Finally, we determined the tradeoff between maximum DNL performance, d_m, and test set size (Figure 9) for DNL test bound in the absence of noise $d'_b = 1.5$ LSB, $\Delta = 0.1$ LSB, and full coverage. Notice that test set size varies inversely with DNL as expected. The average DEC ALPHA CPU times for each data point is 303 seconds.

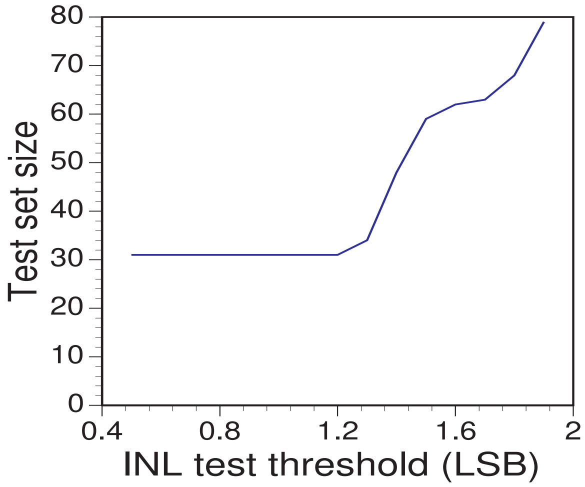

Figure 6: Tradeoff between test set and INL test threshold

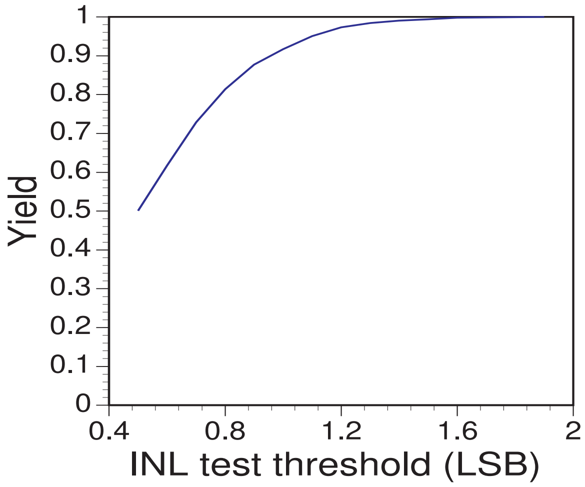

Figure 7: Tradeoff between estimated yield and INL test threshold

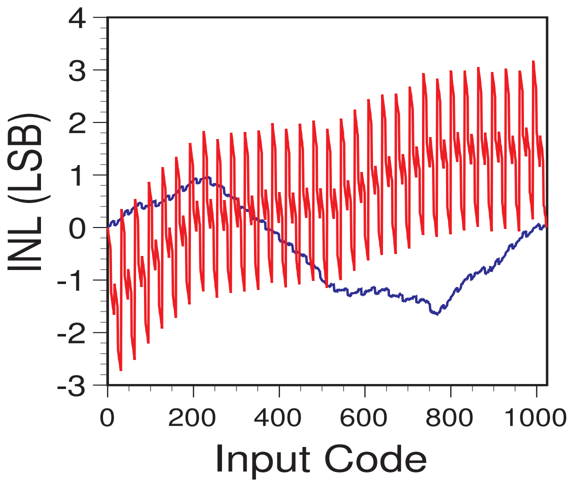

Figure 8: Measured INL of two sample silicon chips

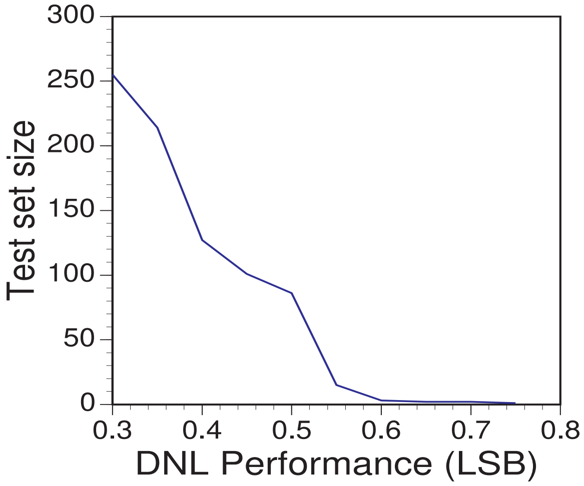

Figure 9: Tradeoff between test set and DNL performance

Next, we applied our strategy to the automatic test generation for an 8 bit cyclic A/D converter. We computed behavioral model parameters from a high level architectural description and expected process variation parameters. In Figure 10, we showed the tradeoff between maximum INL and test set size for $s'_b = 0.8$ LSB, $\varDelta = 0.05$ LSB, and full coverage. Notice that, in general, the test size for this type of A/D converter is very small because the outputs are highly correlated. The average DEC ALPHA CPU times for each data point is 2 seconds.

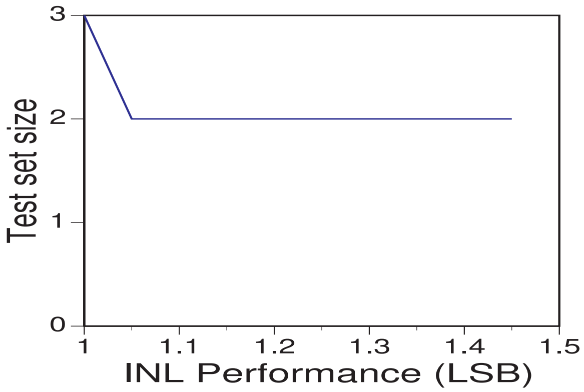

Figure 10: Tradeoff between test set and INL performance

6 Conclusion

We have presented a strategy for testing all DC performance of Nyquist data converters including offset error, full scale gain error, integral nonlinearity, and differential nonlinearity. Assuming that the behavioral model is correct, the testing strategy is exact, but the yield estimate is approximate.

In contrast to previous testing strategies based on linear models that require accurate measurements of circuit performance in the presence of measurement noise, our strategy uses a simpler measurement to verify that a circuit performance parameter falls within a certain range in the presence of measurement noise. Using the proposed strategy and behavioral modeling of the device under test, we evaluated tradeoffs between test set size, detection thresholds, measurement noise, chip performance, and estimated yield. Our results support the obvious that smaller measurement noise, stricter detection thresholds, and lower chip performance would require smaller test set and reduce test time. Stricter detection thresholds, on the other hand, would decrease estimated yield.

References

[1] E. Liu, A. Sangiovanni-Vincentelli, G. Gielen, and P. Gray. A behavioral representation for nyquist rate A/D converters. In *Proc. IEEE ICCAD*, pages 386–389, November 1991.

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

[3] T. Souders and G. Stenbakken. Cutting the high cost of testing. *IEEE Spectrum*, pages 48–51, March 1991.

[4] M. Box and N. Draper. Factorial designs, the $|x'x|$ criterion, and some related matters. *Technometrics*, 13, 1971.

[5] G. Stenbakken and T. Souders. Test-point selection and testability measures via QR factorization of linear models. *IEEE Transactions on Instrumentation and Measurement*, June 1987.

[6] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling. *Numerical Recipes in C. The Art of Scientific Computing*. Cambridge University Press, 1989.

[7] G. Hemink, B. Meijer, and H. Kerkhoff. Testability analysis of analog systems. *IEEE Trans. on CAD*, June 1990.

[8] T. Souders and G. Stenbakken. A comprehensive approach for modeling and testing analog and mixed-signal devices. *IEEE International Test Conference*, 1990.

[9] E. Liu and A. Sangiovanni-Vincentelli. Behavioral simulation for noise in mixed-mode sampled-data systems. In *Proc. IEEE ICCAD*, pages 322–326, Nov 1992.

[10] Analog Devices. *Data converter reference manual volume II*. Norwood, MA 02062-9106, 1992.

[11] W. Lam, A. Saldanha, R. Brayton, and A. Sangiovanni-Vincentelli. Delay fault coverage and performance tradeoffs. *Proc. Design Automation Conference*, June 1993.

[12] 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.

[13] 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.