

Characterization of a Pseudo-Random Testing Technique for Analog and Mixed-Signal Built-In-Self-Test

Jan Arild Tofte¹, Chee-Kian Ong², Jiun-Lang Huang², and Kwang-Ting (Tim) Cheng²

¹Mentor Graphics Corporation, 8005 SW Boeckman Road, Wilsonville, OR, US

²Department of Electrical and Computer Engineering, Univ. of California, Santa Barbara, CA, US
Phone: (503)685-0330, E-mail: tofte@computer.org

Abstract

In this paper, we characterize and evaluate the effectiveness of a pseudo-random-based implicit functional testing technique for analog and mixed-signal circuits. The analog test problem is transformed into the digital domain by embedding the device-under-test (DUT) between a digital-to-analog-converter and an analog-to-digital converter. The pseudo-random testing technique uses band-limited digital white noise (pseudo-random-patterns) as input stimulus. The signature is constructed by computing the cross-correlation between the digitized output response and the pseudo-random input sequence. We have implemented a DSP-based hardware testbed to evaluate the effectiveness of the pseudo-random testing technique. Our test results show that we can achieve close to 100% yield and fault coverages by carefully selecting only two cross-correlation samples. Noise level and total harmonic distortion below 0.1% and 0.5%, respectively, do not affect the classification accuracy.

1. Introduction

The increasing complexity of system-on-chip designs calls for better analog and mixed-signal test methodologies. With only a few analog cores added to a chip design, the cost of testing increases dramatically. It is estimated that by year 2001, more than 60% of application specific integrated circuits (ASIC) will have mixed-signal functions [1]. To overcome the high cost and increasing limitations of external testers, attention has been turned toward BIST (built-in-self-test). Traditionally, analog BIST involves *explicit* functional testing, where the output response is directly measured to see if the performance parameters \mathbf{Z} are within the acceptable range. This approach requires on-chip analog stimuli-generators to produce different types of functional test waveforms. The

signature analysis is performed directly on the analog response, and a large amount of data must be processed. An example of explicit functional testing is spectral BIST, where calibrated sine waves at different frequencies must be generated [2]. Several BIST methodologies rely on various on-chip signal stimuli generators [3–7]. To alleviate some of the problems associated with analog signal generation and signature analysis, a digital test solution was introduced by Ohletz [8]. More recently, the exclusive use of digital test logic to test phase locked loops (PLL) was presented by Sunter and Roy [9]. Since the digital test circuitry is suitable for logic synthesis and automated layout, it is an attractive solution.

A digital pseudo-random testing technique that can be applied to test more general analog circuitry is suggested in [10,11]. The test problem is transformed into the digital domain by embedding the analog circuit components between a digital-to-analog (D/A) converter and an analog-to-digital (A/D) converter to form a digital-in, digital-out module. Figure 1 shows the hardware implementation of the pseudo-random testing technique.

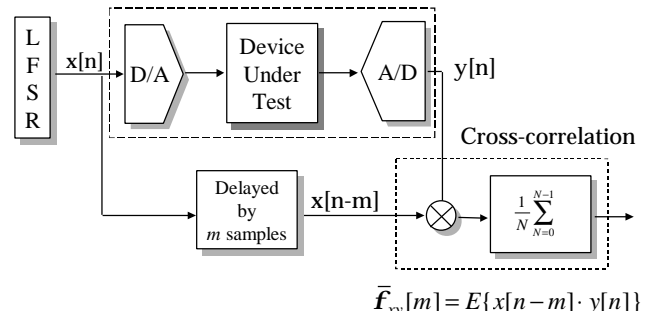


Figure 1. Hardware implementation of the pseudo-random testing technique. The DUT is embedded between the D/A and A/D converter. The estimated cross-correlation instances $\bar{f}_{xy}[m]$ are used as signatures [10].

¹ The work was done when the first author was with Univ. of California at Santa Barbara.

The input stimulus is a sequence of pseudo-random patterns (i.e. band-limited white noise) generated from linear feedback shift registers (LFSR). The signature suggested in [10] is the estimated cross-correlation between the pseudo-random input patterns and the digitized output responses from the DUT. The estimated cross-correlation $\bar{f}_{xy}[m]$ is defined as the expected value of the delayed input sequence $x[n-m]$ and the output sequence $y[n]$ over a period of N patterns:

$$\bar{f}_{xy}[m] = \frac{1}{N} \sum_{n=0}^{N-1} x[n-m] \cdot y[n] \quad (1)$$

The exact computation of $f_{xy}[m]$ requires an infinite number of patterns. A finite number of patterns N are used to estimate the cross-correlation signature samples $\bar{f}_{xy}[m]$. The computational procedure described in Equation 1 is repeated for each signature sample. To compute K estimated signatures, a total of K iterations are required, each with a different delay value for m . The complete signature set \mathbf{f} consists of K signature samples:

$$\mathbf{f} = \{\bar{f}_{xy}[0], \bar{f}_{xy}[1], \bar{f}_{xy}[2], \dots, \bar{f}_{xy}[K-1]\} \quad (2)$$

Figure 2 shows a typical response plot of the signature samples that are produced after cross-correlating the pseudo-random input patterns and the digitized output response of the DUT. The sampling frequency F_s of the D/A and A/D converters determines the time-spacing dT between the signature samples. The significant portion of the response curve is captured by the first K response samples, where K is determined by F_s divided by the 3-dB bandwidth of the DUT:

$$K = \frac{F_s}{BW} \quad (3)$$

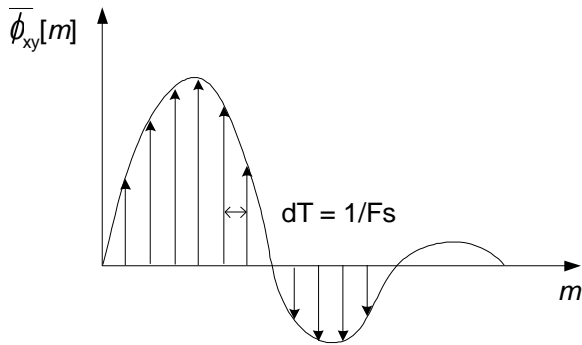


Figure 2. Cross-correlation signature samples. Each value for m gives a signature sample of the response curve.

If the device under test is a linear time-invariant (LTI) circuit, and the input sequence $x[n]$ is stationary white noise with a zero mean value, it can be shown that the cross-correlation $f_{xy}[m]$ is proportional to the impulse response $h[m]$ of the circuit [12]. The impulse response is known to fully characterize a linear, time-invariant system. The estimated cross-correlation instances $\bar{f}_{xy}[m]$ are therefore good signatures for fault detection. This relationship can be derived as follows: The output sequence $y[n]$ of an LTI system with impulse response $h[n]$ is related to the input sequence $x[n]$ by the convolution sum:

$$y[n] = \sum_{k=0}^{\infty} x[n-k] \cdot h[k] \quad (4)$$

The exact cross correlation between the input sequence $x[n]$ and the output sequence $y[n]$ is:

$$f_{xy}[m] = E(x[n] \cdot y[n+m]) \quad (5)$$

Equation 4 can now be substituted for $y[n]$ in Equation 5:

$$f_{xy}[m] = E \left(x[n] \cdot \sum_{k=0}^{\infty} x[n+m-k] \cdot h[k] \right) \quad (6)$$

$$= \sum_{k=0}^{\infty} k[k] \cdot E(x[n] \cdot x[n+m-k]) \quad (7)$$

$$= h[m] \cdot \mathbf{s}_x^2 + \mathbf{m}_x^2 \cdot \sum_{k=0}^{\infty} h[k] \quad (8)$$

If $x[n]$ is stationary white noise, \mathbf{m}_x and \mathbf{s}_x are the mean and standard deviation, respectively [11]. Hence, if the pseudo-random input sequence $x[n]$ has a zero offset voltage, the cross-correlation $f_{xy}[m]$ is proportional to the impulse response $h[m]$.

The technique can be applied to non-LTI circuits as well. Although there is no intuitive interpretation of the signature, it is still likely to be a good signature for fault detection.

The pseudo-random testing technique uses the cross-correlation signature set \mathbf{f} to determine if the DUT is fault-free or faulty. This technique is called *implicit* functional testing since the performance parameters \mathbf{Z} are not directly measured [10]. The parameter set \mathbf{Z} in the performance space is transformed into the signature set \mathbf{f} in the signature space. For an LTI circuit, any changes in \mathbf{Z} are reflected in \mathbf{f} . Fault detection using the signatures is achieved by mapping the known tolerance ranges of the performance parameters into the signature space. The Monte Carlo technique is employed to generate random circuit instances with a known statistical distribution in

performance space. Each circuit instance in the signature space is marked fault-free (faulty) if it satisfies (violates) the tolerance ranges of the performance parameters. The accuracy of this technique depends on how well the fault-free and faulty circuit instances can be separated in the signature space.

In this work, we have implemented a hardware testbed to evaluate and characterize the pseudo-random-based implicit functional testing technique. In particular, we analyze how noise affects the performance of this testing technique.

This paper is organized as follows. The signature analysis and the mapping of the tolerance ranges from the performance space to the signature space are discussed in section 2. Section 3 describes the experimental test procedure and the implementation of the hardware testbed. The test results for a lowpass filter and a bandpass filter are presented in section 4. Concluding remarks, limitations of the pseudo-random testing technique and issues relevant to BIST are addressed in the section 5.

2. Signature Analysis

The mapping of the known tolerance ranges of the performance parameters to the signature space is based on the Monte Carlo method. A large number of circuit instances are generated in the performance space and then mapped into the signature space. The statistical variations of the instances are introduced by changing the device parameter values of the device-under-test. Hence, statistical variations in the *device* space X (e.g. gate widths and resistor values) introduces variations in the performance space Z (e.g. gain and bandwidth). The statistical distributions of the device parameters are processing/layout dependent, and are provided by the manufacturer. It is the statistical uncertainty in the *processing/layout* space P (e.g. precision and energy of the ion beam) that imposes the random disturbance of the device parameters. The distribution of circuit instances in the device, performance, and signature spaces are closely related to the joint probability density function (JPDF) of the statistical disturbance in the processing/layout space, which is usually close to an independent Gaussian distribution [11]. The Monte Carlo induced circuit instances deviates from the nominal instance, and each instance corresponds to a different point in each parameter space. The mapping flow of circuit instances between the parameter spaces is shown in Figure 3.

The circuits used in our experiment are based on operation amplifiers supported by a network of resistors and capacitors. The variations of the device parameters are accomplished by incorporating variable resistors in the circuit designs. The resistor values are changed for each new Monte Carlo run. Each run generates a circuit instance in the performance space that is mapped into the

signature space. If the instance falls within the known performance boundaries in the performance space, it is marked as fault-free in the signature space. If the instance falls outside the performance boundaries, it is considered a parametric fault, and the instance is marked as faulty in the signature space. Recall that a circuit instance in the signature space consists of the complete set of cross-correlation signature samples $f = \{\bar{f}_{xy}[0], \bar{f}_{xy}[1], \bar{f}_{xy}[2], \dots, \bar{f}_{xy}[K-1]\}$. Hence, each signature sample is marked accordingly.

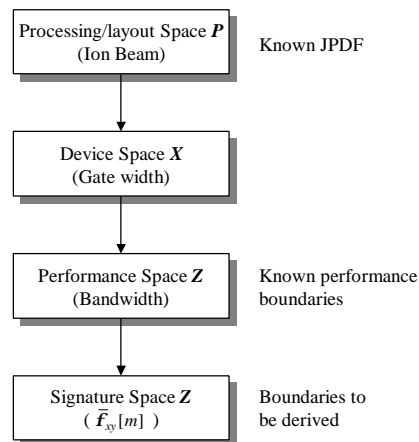


Figure 3. Flow of circuit instance mappings between parameter spaces.

The circuit instances in the signature space are marked as either fault-free or faulty. Our task is to derive boundaries in the signature space such that the misclassification of the instances is minimized. We define the *accuracy* to be the sum of the fault-free and the faulty instances that are classified as fault-free and faulty, respectively, divided by the total instance population. The accuracy gives the percentage of correctly classified instances, and is therefore a good performance metrics. The fault (yield) coverage is defined as the ratio of circuit instances that are correctly classified as faulty (fault-free) to the total number of faulty (fault-free) circuit instances.

The signature set f of each circuit instance consists of K cross-correlation signature samples. Potentially, the classification problem entails deriving discriminating boundaries in the K -dimensional signature space. However, our experiments show that very high classification accuracy can be achieved by carefully selecting only two cross-correlation signature samples. We refer to these two selected samples as a *signature-pair*. Hence, the classification problem is reduced to deriving the boundaries in a 2-dimensional signature space. The boundaries consist of a polygon derived from a convex hull function that encloses all fault-free instances in the training-set (100% yield coverage). The fault

coverage and classification accuracy depend on how many faulty instances in the evaluation-set are wrongly classified because they end up at the wrong side of the polygon boundaries.

The decision of using a polygon to discriminate the fault-free from the faulty circuit instances is based on the assumption that the fault-free circuit instances tend to cluster together. The performance requirements for a circuit are specified such that the parameters are within an acceptable range of values. Our experiments show that the acceptable range in the performance space translates into a cluster of circuit instances in the K -dimensional signature space. The signature space consists of one fault-free instance-cluster surrounded by a cloud of faulty instances. The fault-free instance-cluster can also be found in the 2-dimensional signature space, if the signature-pair is carefully selected. An exhaustive search is performed to find the best signature-pair that gives the highest classification accuracy.

Good discrimination boundaries are achieved by first considering a *training-set* of circuit instances. For the experimental procedure discussed in section 3, effort is made to generate circuit instances that are very close to the performance boundaries. After the polygon is derived, an *evaluation-set* of circuit instances is generated. The instances in the *evaluation-set* should be generated based on the statistics in the processing/layout space, and follow a normal distribution with respect to the nominal circuit instance. Due to the manual effort of changing the resistor values between each Monte Carlo run, the circuit instances in our experiments follow a uniform distribution. The instances of the evaluation-set are classified as fault-free (faulty) if they fall within (outside) the fault-free region established by the polygon. The misclassification is caused by 1) mixing of fault-free and faulty instances in the signature space, and 2) the uncertainty of the estimated signatures [10].

3. Testbed and Experimental Procedure

3.1. Experimental Setup

The experimental testbed used to evaluate and characterize the pseudo-random testing technique is shown in Figure 4. The hardware implementation of the testbed consists of two high performance data acquisition boards from Gage Applied Sciences. A 12-bit D/A board (CompuGen 1100) operating as an arbitrary waveform generator converts the digital pseudo-random test patterns into band-limited analog white noise. Similarly, a 12-bit A/D board (CompuScope 8012) captures the response from the analog device-under-test. The D/A and A/D boards in the testbed correspond to the D/A and A/D converters in a practical BIST solution. The boards are

interfaced through the ISA bus of a 300 MHz Dell Pentium II PC.

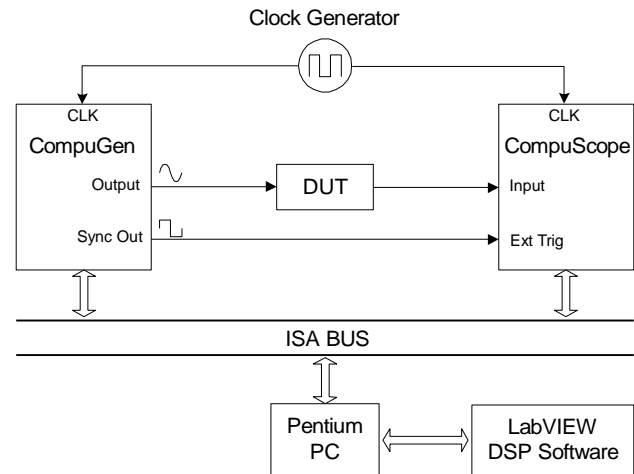


Figure 4. Experimental testbed implementing the pseudo-random testing technique.

The 12-bit pseudo-random patterns are generated in software and loaded into CompuGen's on-board memory-buffer. A *test-session* is defined as the delivery of one complete test sequence of pseudo-random patterns to the DUT, and the consequent capture of the response from the DUT. The analog test waveform generated at the output of CompuGen corresponds to the sequence of pseudo-random digital patterns residing in the memory-buffer. Similarly, CompuScope captures and stores the sampled response from the DUT in its memory-buffer. The captured response is transferred to the application software for post-processing and analysis. All programming is done in LabVIEW, which is an efficient programming environment for measurements, control, and signal processing. All signal connections shown in Figure 4 are accomplished by 50Ω coaxial cables.

A maximum sampling frequency of 80 MHz for both boards allows us to test analog circuits with bandwidths up to 40 MHz. To ensure that CompuGen and CompuScope are synchronized, a HP8116A function generator provides both boards with the same external sampling clock. CompuGen and CompuScope have memory-buffers of 512k and 1M data samples, respectively. In our setup, CompuGen is configured to operate in a continuous mode. After outputting the last sample in the pseudo-random sequence, the memory-buffer address-pointer wraps around to the first sample again. The result is a continuous sequence of test-sessions, where each test-session consists of a maximum of 512k pseudo-random patterns. In the same way as CompuGen is configured to continuously apply test-sessions to the DUT, CompuScope's memory-buffer is

configured as a circular buffer that wraps around and overwrites its previously captured test-sessions. At the beginning of each test-session CompuScope's *Ext Trig* input receives a trigger signal from the *Sync Out* terminal of CompuGen. *Sync Out* is a TTL signal programmed to pulse high at the beginning of each new test-session, i.e. at the same time as the sample clock triggers the first sample. When CompuScope receives a trigger signal, it marks the current sample address in its memory-buffer as the starting point of the test-session. After the desired number of samples has been captured, the samples are transferred to the application software for post-analysis.

To summarize, the sequence of events to complete a test-session is as follows: 1) load the pseudo-random patterns into CompuGen's memory-buffer, 2) instruct CompuGen to start continuous delivery of test-sessions, 3) arm CompuScope to hardware trigger on CompuGen's *Sync Out* signal, 4) when a trigger is detected, mark the starting point of the test-session, and 5) when the current test-session has been successfully captured, transfer the response samples from CompuScope's memory-buffer to the application software. There is a certain software delay from the point CompuGen is instructed to start delivering test-sessions until CompuScope is armed for triggering. Several test-sessions can pass before CompuScope successfully triggers on a test-session. If the same test pattern will be used for several captures, steps 3 through 5 can be repeated without involving steps 1 and 2. When the application software is ready for another capture, CompuScope is simply rearmed to trigger on another test-session.

3.2. Calibration of Pattern Generator

After the response is transferred to the application software it is cross-correlated with the applied pseudo-random test pattern. In the BIST implementation of Figure 1, the test pattern is taken directly from the LFSR. Similarly, in the testbed implementation the test pattern is already available in the application software where the pseudo-random test sequence was initially generated. If the output of CompuGen is connected directly to the input of CompuScope—i.e. bypassing the DUT—the sampled response is not exactly the same as the applied pseudo-random test sequence. In particular, the sampled response does not reach the maximum amplitude of the applied signal. If the output range of CompuGen and the input range of CompuScope are both set to an amplitude of 0.5 volts, the 12-bit response samples should reflect the same maximum amplitude in terms of a numeric range from -2047 to $+2048$. Depending on the amplitude range, the response reaches its maximum around ± 2000 . This discrepancy is primarily caused by the output amplifiers of CompuGen, which are tuned to satisfy a particular voltage range for either a 50Ω or $1M\Omega$ load impedance.

Both CompuScope and the DUT are terminated at 50Ω , but CompuGen's sensitivity to load impedance results in slightly different amplitude levels. There is also a small offset voltage present at CompuGen's output. Because of these non-ideal factors, we do not use the software generated pseudo-random patterns when doing the cross-correlation. Instead, we bypass the DUT and use the captured response created by the *ideal* pseudo-random patterns. This *calibrated* signal is closer to what is actually applied to the DUT. Henceforth, before starting experimental testing of a circuit, we perform a test-session calibration run where the DUT is disconnected. The coax input to the DUT is directly connected to the coax output of the DUT. Furthermore, during calibration, a constant is added to the pseudo-random test patterns to eliminate any DC offset. A few calibration runs are necessary before the adjustment of the constant results in a zero DC offset. The maximum amplitude of the pseudo-random patterns is reduced by the absolute value of the same constant to prevent the test signal from exceeding the -2047 to $+2048$ numeric range. The average of 25 test-sessions gives the final calibrated pseudo-random test pattern.

During a calibration run when the DUT is bypassed, CompuScope receives the TTL trigger signal a few samples before the analog test waveform arrives. If the analog signal is a square wave, it can be observed that the captured square wave lags behind the applied square wave. Specifically, the delay is measured to be 17 samples. This delay problem is compensated for by advancing the address start-pointer for CompuScope's memory-buffer 17 samples. This delay is constant for the range of sample frequencies used in our experiments.

No special effort has been devoted to reduce the inherent noise of the testbed and the surrounding environment. Ideally, the data acquisition boards would be installed in a low-noise instrumentation PC. The DUT is not shielded from external electromagnetic noise sources, but all signal connections to the boards and the DUT are accomplished through shielded coaxial cables. The input to CompuScope is equipped with an anti-aliasing filter to ensure that the frequency components above the Nyquist frequency are attenuated. To increase the signal-to-noise (S/N) ratio, the amplitude range for CompuGen and CompuScope are selected to closely match the maximum input and output level specifications of the DUT, respectively. For the experiments presented in this paper, the output amplitude range for CompuGen is set to 0.4 volts, and the input voltage range for CompuScope is set to 0.5 volts. Hence, the DUT can have a maximum gain of 1.25 (1.94 dB) before CompuScope starts clipping the signal. As will be discussed in section 4, the noise level with the embedded DUT is around 0.1 percent, which is the equivalent of a S/N ratio of 60 dB. The total harmonic distortion (THD) is around 0.3 percent.

3.3. Evaluating Circuit Performance Parameters

In addition to computing the cross-correlation signature, the application software also performs the required performance testing of the DUT. For each Monte Carlo test-session, the cross-correlation signatures are obtained. These signatures are marked fault-free (faulty) if the DUT satisfies (violates) its performance constraints. The performance parameters considered are the DC offset, and the gain and phase spectrum. The pseudo-random band-limited white noise that is applied to the DUT for the purpose of obtaining the cross-correlation can also be used to compute the gain and phase spectrum. The procedure is as follows: 1) from the white-noise test stimulus and response, compute the impulse response, 2) take the FFT of the impulse response to get the transfer function of the DUT, and 3) extract the gain and phase parameters from the transfer function. The accumulative effect of the sequential steps above results in a high-frequency noise disturbance of the final output. The phase spectrum is therefore conditioned by the DSP implementation of a 3rd order Butterworth filter. Although this approach proves to be acceptable for the phase spectrum, the gain spectrum is plagued by additional noise in the lower end of the frequency spectrum. We therefore use a general frequency-sweep approach to obtain the gain of the DUT. A series of test-sessions are performed where CompuGen is configured to generate sine waves. Each test-session targets a different frequency. Enough frequency points are selected to obtain a good measurement of the complete gain spectrum. For each test-session, the AC amplitude of the captured sine wave is calculated. Similarly, the AC amplitude in the absence of the DUT is recorded during a sine-wave calibration run. Since the calibrated AC amplitude is the same throughout the entire spectrum, only one frequency point is required. The gain spectrum is calculated by dividing the AC amplitudes for the different frequency points with the calibrated AC amplitude. The DC offset of the circuit is also calculated using the frequency-sweep approach, but this value is not related to CompuGen's calibrated DC offset that is close to zero.

3.4. Field Programmable Analog Device as DUT

In our experiments, the DUT is realized using a field programmable analog device (FPAD) from Zetex Semiconductors. The TRAC020 is a totally reconfigurable analog circuit (TRAC) with 20 programmable analog cells. The TRAC development board has four TRAC020 chips, but one chip is often enough to implement analog designs. A PC application software downloads the circuit configuration to the development board through the parallel port. Unlike many other FPADs that are capacitor switched, the datapath in the TRAC020 is purely analog. The bandwidth is typically 4 MHz for a 10 mV signal.

Unfortunately, the bandwidth is only around 450 kHz for the 0.4 V amplitude range used in our experiments. Some of the analog functions that can be implemented in the TRAC cells rely on external resistors and capacitors. The pins of the TRAC020 chips are connected to sockets on the development board. The external components are plugged into these sockets. In our experiments, we change the values of variable resistors to simulate parameter variations in the DUT. Variable capacitors are not used since they are small and hard to adjust. Only changing the resistor values easily alters the RC constants in our experimental circuits. The resistors are incrementally programmable cermet potentiometers from Bourns (part# 3683). The range is from 0 to 5 k Ω , and the resolution is 0.1% (5 Ω).

Different circuit instances of the DUT are generated by changing the value of only one resistor for each new Monte Carlo test-session. After adjusting one resistor through a range of values, it's set back to its nominal value before moving on to the next resistor—i.e. we only consider single-faults.

To summarize, the experimental steps following the calibration procedure are: 1) adjust the value of one external resistor, 2) obtain the DC offset and gain spectrum by applying a series of frequency-sweep test-sessions, 3) perform a test-session based on the pseudo-random test pattern to obtain the phase spectrum and cross-correlation, 4) check if the circuit instance of the DUT satisfies/violates the performance parameters, and 5) mark the cross-correlation signature set fault-free/faulty, and log the results to file. These steps are repeated for each new Monte Carlo test-session.

The testbed implements the signature analysis part in software, but since the computational procedure is DSP-based, a realistic hardware environment for the device-under-test is maintained.

4. Test Results

In this section we present experimental results for two circuits—a lowpass filter and a bandpass filter, which are shown in Figures 5a and 5b, respectively.

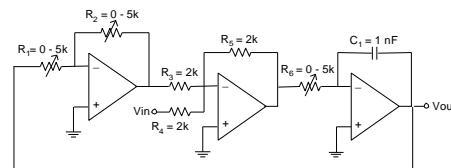


Figure 5a. Lowpass filter.

Performance parameters for the lowpass filter:

- * Pass-band gain (0 kHz – 280 kHz): $-4.5 \text{ dB} \leq A_0 \leq -0.5 \text{ dB}$.
- * Stop-band attenuation: $A_0 \leq -9 \text{ dB}$ at 400 kHz.
- * Pass-band phase margin: $\phi \leq 5^\circ$.
- * DC offset in pass-band: $V_{os} \leq 2.4 \text{ mV}$.

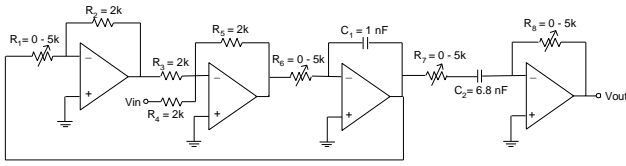


Figure 5b. Bandpass filter.

Performance parameters for the bandpass filter:

- * Pass-band gain (50 kHz – 200 kHz): $-1.5 \text{ dB} \leq A_0 \leq 1.5 \text{ dB}$.
- * Stop-band attenuation: $A_0 \leq -8 \text{ dB}$ at 10 kHz and 200 kHz.
- * Pass-band phase margin: $\phi \leq 5^\circ$.
- * DC offset in pass-band: $V_{os} \leq 2.4 \text{ mV}$.

The performance constraints result in the fault-free ranges for the resistor values as shown in Tables 1a and 1b. The nominal resistor values represent the nominal circuit instance.

Table 1a. Nominal and fault-free ranges of the variable resistors for the lowpass filter.

	Nominal	Fault-Free Range
R1	2500Ω	1350Ω – 2790Ω
R2	2925Ω	2615Ω – 3105Ω
R6	1150Ω	1005Ω – 1220Ω

Table 1b. Nominal and fault-free ranges of the variable resistors for the bandpass filter.

	Nominal	Fault-Free Range
R1	2250Ω	1935Ω – 2435Ω
R6	825Ω	740Ω – 990Ω
R7	2925Ω	2605Ω – 3405Ω
R8	2875Ω	2510Ω – 3160Ω

For both circuits, we characterize how noise affects the signature stability. A common technique in data acquisition for reducing the noise level is to average several captures. For each Monte Carlo circuit instance, we average different numbers of captured test-sessions. Assuming a uniform noise distribution, the noise is

reduced by a factor of $\frac{1}{\sqrt{n}}$, where n is the number of

averaged captures. We investigate five different values for n —1, 4, 9, 16, and 25. To measure how different values for n relates to the actual noise and total harmonic distortion (THD), we perform spectral analysis of the captured response. The sampling rate for CompuGen and CompuScope is set to 20 MHz, and each test-session consists of 32k patterns. The resistors are set to their

nominal values, and a sine wave is used as test stimulus. The fundamental frequencies used for the lowpass and bandpass filters are 20.625 kHz (33 cycles) and 128.75 kHz (206 cycles), respectively. All harmonics within the bandwidth of the circuits are considered for noise analysis. The mean of ten different noise measurements is used. As the results in Table 2 indicate, the noise level

closely follows a reduction of $\frac{1}{\sqrt{n}}$, where n is the number of averaged captures. The THD is as expected not affected by the number of averaged captures.

Table 2. Noise and THD for the lowpass and bandpass filters, where n is the number of test-session captures that has been averaged.

n	Lowpass Filter		Bandpass Filter	
	Noise (%)	THD (%)	Noise (%)	THD (%)
1	0.1368	0.1939	0.0695	0.4882
4	0.0477	0.1898	0.0259	0.4882
9	0.0293	0.1916	0.0172	0.4882
16	0.0220	0.1899	0.0150	0.4881
25	0.0186	0.1907	0.0129	0.4885

The performance evaluation and the pseudo-random testing of the circuits are also performed using a sample frequency of 20 MHz and 32k patterns per test-session. To ensure that the significant portion of the response is captured, the first 100 cross-correlation signature points are recorded. Experimental data based on 16 averaged captures is presented in Figures 6 through 8. Figures 6a and 6b show the cross-correlation response-curves for the lowpass and bandpass filters, respectively. A few randomly selected circuit instances are plotted. Note that the inverted output of the bandpass filter results in a negative cross-correlation response.

Figures 7a and 7b show the circuit instance signatures of the training-set for the lowpass and bandpass filters, respectively. The signature-pair with time delays 34 and 69 is used for the lowpass filter, and the signature-pair with time delays 28 and 53 is used for the bandpass filter. The resistor values are changed with 5 Ω increments (0.1%) for the first 3 Monte Carlo samples at each side of the boundaries. The actual boundary violations in the performance space for such a small change is only about 0.01 – 0.02 dB. Hence, the cross-correlation response is very sensitive to small changes in the performance space. A total of 19 fault-free and 19 faulty circuit instances were generated for the lowpass filter. For the bandpass filter, a total of 43 fault-free and 43 faulty circuit instances were generated. The polygon is derived to discriminate between the fault-free and the faulty circuit instances in the training-set.

Figures 8a and 8b show the circuit instance signatures of the evaluation-set for the lowpass and bandpass filters, respectively. A total of 23 fault-free and 23 faulty circuit instances were generated for the lowpass filter. For the bandpass filter, a total of 32 fault-free and 32 faulty circuit instances were generated. Note that circuit instances related to the change of one particular resistor exhibit a linear trend in the 2-dimensional signature space. This is because the change in the cross-correlation magnitude for the selected instance samples are linearly proportional to each other. If a different signature-pair is selected, the 2-dimensional plot looks completely different, and in most cases, the fault-free and faulty instances are intermixed.

The polygon derived for the training-set is applied to classify the circuit instances in the evaluation-set. As figures 8a and 8b indicate, high classification accuracy is achieved for the selected signature-pairs. For the bandpass filter 100% classification accuracy is achieved using signature-pair (28,53). There exists several other signature-pairs that also gives 100% classification accuracy. For the lowpass filter, two fault-free instances escape the boundaries of the polygon, and are therefore wrongly classified as faulty. This leads to 96% classification accuracy, 100% yield coverage, and 92% fault coverage for the lowpass filter using signature-pair (34,69). However, there exists other signature-pairs that result in 100% classification accuracy for the lowpass filter as well.

The boundary derivation from the training-set, and the classification assessment of evaluation-set are performed for all possible signature-pairs. The signature-pairs resulting in the highest classification accuracy are preferred. Several of the 4950 possible signature-pairs result in 100% classification accuracy for the lowpass and bandpass filters. Furthermore, the noise does not seem to affect the classification accuracy. Table 3 shows the percentage of signature-pairs that result in 100% classification accuracy for the different numbers of averaged captures n .

Table 3. Percentage of signature-pairs that result in 100% classification accuracy.

n	Signature-pairs (%)	
	Lowpass Filter	Bandpass Filter
1	1.43	7.17
4	1.54	7.21
9	1.73	6.62
16	1.66	5.68
25	1.82	6.46

If the desired classification accuracy is reduces by a few percent, the number of signature-pairs increases dramatically. This behavior is indicated in Table 4—using the case where n is 9. This trend is also consistent with different numbers for n —indicating that the noise does not affect the classification accuracy.

Table 4. Percentage of signature-pairs that ensures a given classification accuracy

Accuracy (%)	Signature-pairs (%)	
	Lowpass Filter	Bandpass Filter
100	1.73	6.62
99	1.73	6.62
98	1.73	31.72
97	7.01	31.72
95	14.28	54.20
90	24.69	66.73

The classification methodology favors yield coverage since the polygon is derived such that all the fault-free circuit instances in the training-set are inside the boundaries of the polygon. However, there exists a complementary method that favors fault coverage. With the same assumption, this classification is based on deriving a polygon such that all the faulty circuit instances are outside the boundaries of the polygon.

Figure 6a. Lowpass Filter.
Cross-Correlation Response Curves.

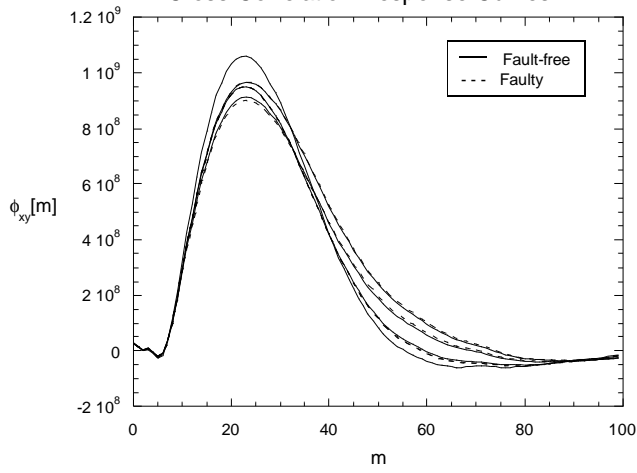


Figure 6b. Bandpass Filter.
Cross-Correlation Response Curves.

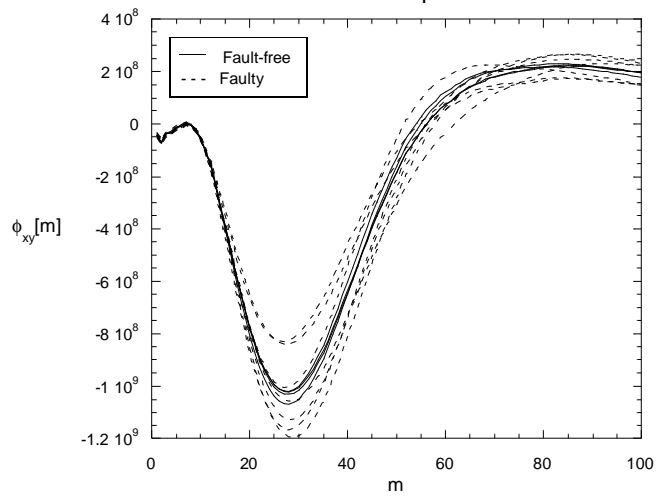


Figure 7a. Lowpass Filter.
Defining Boundary From Training Set.

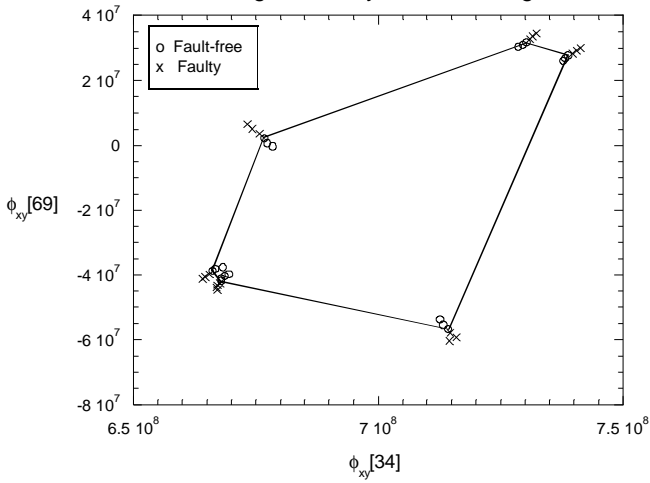


Figure 7b. Bandpass Filter.
Defining Boundary From Training Set.

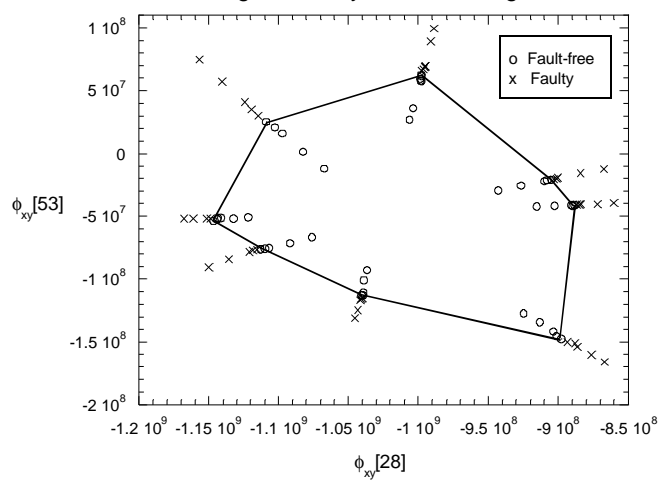


Figure 8a. Lowpass Filter.
Classification of Training Set.

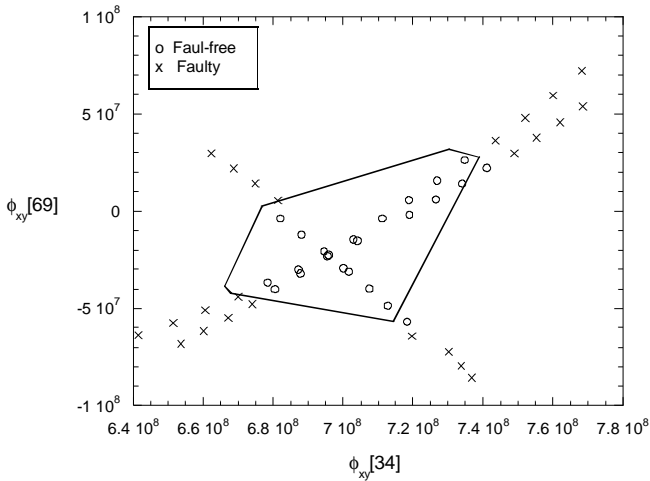
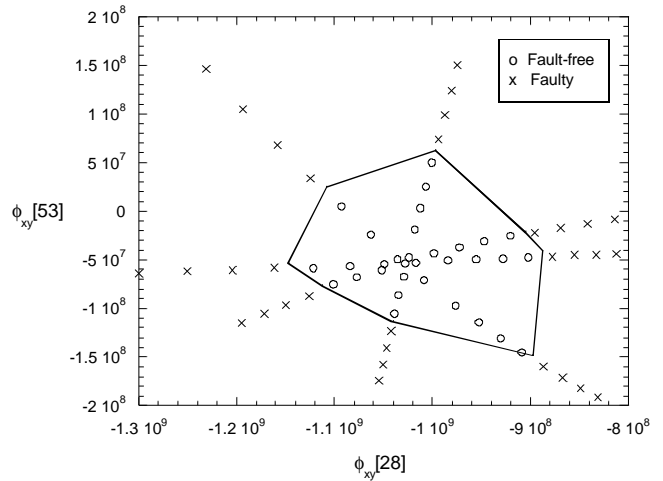


Figure 8b. Bandpass Filter.
Classification of Training Set.



5. Conclusion

In this paper, we described a hardware testbed implementation of a pseudo-random testing technique. Our experimental results show that the cross-correlation signatures are very sensitive to variations in the performance parameters, and are capable of detecting performance fluctuations of 0.01dB. The fault-free and faulty instances are clearly separated in the signature space. We presented a classification method that effectively discriminates between the fault-free and faulty circuit instances in the signature space. For the lowpass and bandpass filters, there exists several signature-pairs that achieve 100% classification accuracy. More complex circuits may not be suitable for this kind of classification. This may require selecting more than two cross-correlation signatures to achieve the desired classification accuracy.

Ideally, a larger number of generated circuit instances for the training-set and evaluation-set is desired. The manual effort of changing one of the resistor values for each Monte Carlo run, limits the number of instances that can be generated in a timely manner. Programming the resistors directly from the computer would eliminate the need for human interaction during simulation.

The primary target for the pseudo-random testing technique is BIST. The noise level should be far less for an on-chip environment than the numbers reported for our testbed implementation. Many chips make use of DSP cores where D/A and A/D converters are available. Implementation of the pseudo-random testing technique for BIST should therefore be attractive in terms of low area overhead. It might be too expensive to implement this testing technique if either D/A or A/D converter is not available on the chip. Another limiting factor for BIST implementation is the clock speed of the D/A and A/D converters. The sample frequency should be at least twice the bandwidth of the analog circuit components. Testing of analog components in the RF range is therefore currently not feasible.

With the exception of the D/A and A/D converters, the required on-chip test hardware is very efficient. Our test results show that only two (or possibly a few more) cross-correlation signatures need to be computed if the time delays for the signature-pairs are carefully selected. This helps speeding up the on-chip processing time. If each cross-correlation signature is represented by n bits, the signatures can be serially unloaded from an n -bit shift register. This assumes that the hardware implementation is such that one cross-correlation signature is unloaded while the next one is being calculated. We call this shift register a cross-correlation signature register (CCSR). In the same way as the signature in digital BIST is captured in a multiple input shift register (MISR), the signature for

the pseudo-random testing technique is captured in the CCSR. The comparative analysis performed by the ATE involves verifying that the 2-dimensional, or possibly multi-dimensional representation of the signatures falls within the discriminating boundaries of the polygon.

Unlike other analog and mixed-signal test solutions, the pseudo-random testing technique combined with our proposed classification method is extremely efficient in terms of I/O bandwidth. The on-chip signal generation tests the entire frequency spectrum of the circuit without any input waveform requirements. Secondly, the digital test response from the CCSR can be shifted off-chip using only one output pin.

The mixed-signal aspect of this test solution comes from the possibility of sharing the test hardware between the analog and digital test-sessions. Digital BIST can make use of the same LFSR that is used to generate band-limited white noise. The CCSR can be reconfigured as a MISR when testing the digital circuitry.

References

- [1] S. Sunter, "IC Techniques for Mixed-Signal DFT and BIST," Tutorial 15, *Int. Test Conf.*, November 1999.
- [2] B. Vinnakota, *Analog and Mixed-Signal Test*. 1998, NJ: Prentice-Hall Inc.
- [3] M. F. Toner and G.W. Roberts, "A BIST scheme for a SNR, gain tracking, and frequency response test of a sigma-delta ADC," *IEEE Transactions on Circuits and Systems*, vol 42, pp. 1-15, Jan. 1995.
- [4] K. Arabi, B. Kaminska, and J. Rzeszut, "A new Built-In Self-Test approach for digital-to-analog and analog-to-digital converters," *Proc. Int. Conf. Computer-Aided Design*, Nov. 1994, pp. 491-494.
- [5] M. Ehsanian, B. Kaminska, and K. Arabi, "A New Digital Test Approach for Analog-to-Digital Converter Testing", *14th VLSI Test Symposium*, 1996, pp. 60-65.
- [6] E. Teraoka *et al.*, "A Built-In Self-Test for ADC and DAC in a Single-Chip Speech CODEC," *Proc. Int. Test Conf.*, Oct 1993, pp. 791-796.
- [7] B. Provost, E. Sanchez-Sinencio, "Auto-Calibrating Analog Timer for On-Chip Testing," *Proc. Int. Test Conf.*, Sept 1999, pp. 541-548.
- [8] M. Ohletz, "Hybrid Built-In Self-Test for Mixed Analog/Digital Integrated Circuits", *Proc. European Test Conference*, April 1991, pp. 307-316.
- [9] S. Sunter and A. Roy, "BIST for Phase-Locked Loops in Digital Applications," *Proc. Int. Test Conf.*, Sept 1999, pp. 532-540.
- [10] C.Y. Pan and K.T. Cheng, "Implicit Functional Testing for Mixed-Signal Circuits", *Proc. VLSI Test Symposium*, April 1996, pp. 489-494.
- [11] C.Y. Pan and K.T. Cheng, "Pseudo-Random Testing for Mixed-Signal Circuit", *IEEE Trans. on CAD*, Oct. 1997, pp. 1173-1185.
- [12] C.W. Therrien, *Discrete Random Signals and Statistical Signal Processing*. 1992: Prentice-Hall Inc.