Southern Illinois University Carbondale
OpenSIUC

Articles

Department of Electrical and Computer Engineering

10-2005

A Methodology to Perform Online Self-Testing for Field-Programmable Analog Array Circuits

Amit Laknaur
Southern Illinois University Carbondale

Haibo Wang
Southern Illinois University Carbondale, haibo@engr.siu.edu

Follow this and additional works at: http://opensiuc.lib.siu.edu/ece_articles
Published in Laknaur, A., & Wang, H. (2005). A methodology to perform online self-testing for field-programmable analog array circuits. IEEE Transactions on Instrumentation and Measurement, 54(5), 1751 - 1760. doi: 10.1109/TIM.2005.855096 ©2005 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE. This material is presented to ensure timely dissemination of scholarly and technical work. Copyright and all rights therein are retained by authors or by other copyright holders. All persons copying this information are expected to adhere to the terms and constraints invoked by each author's copyright. In most cases, these works may not be reposted without the explicit permission of the copyright holder.

Recommended Citation
Laknaur, Amit and Wang, Haibo, "A Methodology to Perform Online Self-Testing for Field-Programmable Analog Array Circuits" (2005). *Articles.* Paper 9.
http://opensiuc.lib.siu.edu/ece_articles/9

This Article is brought to you for free and open access by the Department of Electrical and Computer Engineering at OpenSIUC. It has been accepted for inclusion in Articles by an authorized administrator of OpenSIUC. For more information, please contact opensiuc@lib.siu.edu.

A Methodology to Perform Online Self-Testing for Field-Programmable Analog Array Circuits

Amit Laknaur and Haibo Wang

***Abstract*—This paper presents a methodology to perform online self-testing for analog circuits implemented on field-programmable analog arrays (FPAAs). It proposes to partition the FPAA circuit under test into subcircuits. Each subcircuit is tested by replicating the subcircuit with programmable resources on the FPAA chip, and comparing the outputs of the subcircuit and its replication. To effectively implement the proposed methodology, this paper proposes a simple circuit partition method and develops techniques to address circuit stability problems that are often encountered in the proposed testing method. Furthermore, error sources in the proposed testing circuit are studied and methods to improve the accuracy of testing results are presented. Finally, experimental results are presented to demonstrate the validity of the proposed methodology.**

***Index Terms*—Analog online testing, built-in-self-testing (BIST), field-programmable analog array (FPAA).**

I. Introduction

FIELD-PROGRAMMABLE analog arrays (FPAAs) are the counterparts of field programmable gate arrays (FPGAs) in analog domain. In the past few years, numerous efforts have been devoted to developing FPAA technologies [1]–[3]. At present, quite a few commercial FPAA products are already on the market.[1,2,3] Such devices have been used to implement signal conditioning, filtering, and other analog functions for a wide range of applications. As FPAAs can be reconfigured in the field, they are very promising for implementing analog circuits with self-testing and self-repairing capabilities. Such circuits are extremely desirable in military, space, and certain commercial applications, in which electronic systems operate in harsh environments without easy access for maintenance. Although promising experiments have been reported on conducting fault-recovery operations on FPAA circuits [4], the design of self-testing and self-repairing analog circuits on FPAA platforms remains an open question. One obstacle to FPAA self-repairing circuits is the lack of techniques to perform self-testing and consequently locate faulty blocks on FPAA circuits. To address this challenge, this paper presents an efficient online testing method for FPAA circuits. The developed method is also capable of locating faulty FPAA blocks, making it suitable for the design of self-repairing FPAA circuits.

Manuscript received August 12, 2004; revised April 29, 2005.

The authors are with the Department of Electrical and Computer Engineering Southern Illinois University, Carbondale, IL 62901 USA.

Digital Object Identifier 10.1109/TIM.2005.855096

[1]Anadign Inc. Available: http://www.anadigm.com

[2]Cypress Semiconductor. Available: http://www.cypress.com

[3]Lattice Semiconductor. Available: http://www.latticesemi.com

Several FPAA testing techniques have previously been developed [5]–[8]. However, these techniques mainly focus on FPAA offline testing. In this paper, we refer to *online* testing as a testing process that does not affect the normal operation of the circuit under test (CUT). On the contrary, *offline* testing needs to halt the normal operation of the CUT for testing purposes. For general analog circuits, online testing is conducted using either *redundancy-based* or *nonredundancy-based* approach. The redundancy-based approach duplicates a portion or the entire CUT, and compares the outputs of the original circuit and its replication [9]–[11]. The nonredundancy-based approach relies on built-in-self-testing (BIST) circuits to measure certain performance metrics of the CUT, such as current consumption [12], statistical properties of circuit characteristics [13], common-mode signal levels in fully differential circuits [14]–[16], and other types of signal values [17]–[21]. By taking advantage of special properties of the CUT, the nonredundancy-based approach normally requires small hardware overhead. However, this approach sometimes involves testing circuits that are difficult to design. On the contrary, the redundancy-based approach usually results in significant hardware cost. Yet it is easy to implement and, more interestingly, can be used to locate malfunctioning circuit blocks.

The proposed testing methodology takes the redundancy-based approach. It partitions the CUT into subcircuits. To test a subcircuit, programmable resources on the FPAA chip are configured to replicate the subcircuit. The subcircuit under test and its replication are fed with the same inputs, and their outputs are compared to check if faults have occurred in the circuit. Since the same FPAA programmable resources can be reconfigured to duplicate different subcircuits, this approach results in small hardware overhead and provides additional flexibility. To efficiently implement the proposed methodology, this paper presents a simple partition method for FPAA circuits. Also, it develops circuit techniques to address stability problems, which are often encountered in the redundancy-based online testing approach. Furthermore, error sources in the proposed testing circuits are studied, and methods to improve the accuracy of testing results are presented. Finally, experimental results are presented to demonstrate the feasibility of the proposed testing methodology.

The rest of the paper is organized as follows. Section II gives a brief introduction to FPAA technologies. In Section III, the proposed online testing methodology is described. Section IV discusses the partition method for FPAA circuits. Techniques to solve circuit stability problems and methods to improve testing accuracy are presented in Section V. Experimental results are presented in Section VI, and the paper is concluded in Section VII.

0018-9456/$20.00 © 2005 IEEE

Authorized licensed use limited to: Southern Illinois University Carbondale. Downloaded on May 28, 2009 at 17:36 from IEEE Xplore. Restrictions apply.

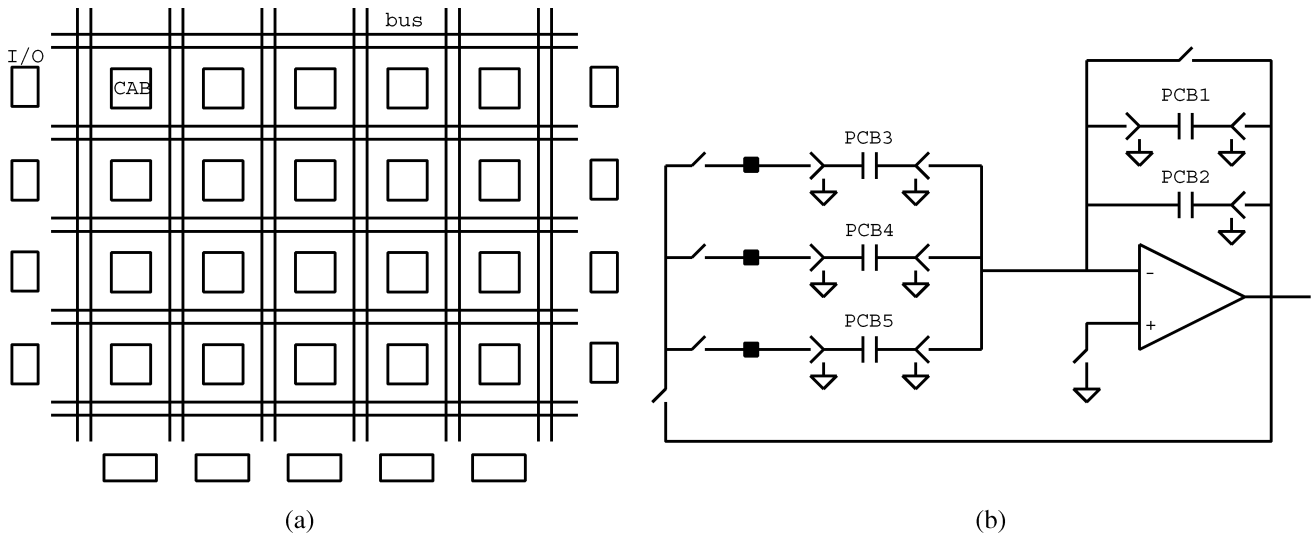


Fig. 1. FPAA structure [22]. (a) FPAA layout. (b) FPAA CAB structure.

II. FPAA Technologies

An FPAA device typically contains configurable analog blocks (CABs), interconnect networks, I/O circuits, and on-chip memories. CABs consist of primitive analog components whose values and connections can be programmed to implement simple analog functions. Programmable interconnects route signals around CABs to realize more sophisticated analog functions. I/O circuits provide interface between FPAA internal circuits and outside systems. Finally, on-chip memories store FPAA configuration bitstreams.

Among various circuit techniques to implement FPAA circuits [1], switched-capacitor (SC) technology is particularly attractive to the design of FPAAs, mainly because SC circuits have high accuracy, are insensitive to parasitics, and can be easily programmed. As an example, Fig. 1 shows an SC-based FPAA architecture [22], [23], which was produced by Motorola. A modified version of this FPAA architecture is currently produced by Anadigm Inc. As shown in Fig. 1, this architecture contains 20 CABs, arranged into a 4×5 array. Around the three sides of the CAB array, there are 13 I/O circuits. Each CAB consists of an operational amplifier (op-amp), five programmable capacitor banks, and a number of switches. The values of capacitor banks as well as the states of the switches can be configured to implement different functions. Since SC circuits are discrete-time circuits by nature, exact circuit analysis for SC-based FPAAs should be performed in the z domain. However, when signal frequencies are significantly smaller than (e.g. smaller than one-tenth of) the clock frequencies of SC circuits, FPAA circuits can be approximately treated as continuous-time circuits and, hence, their operations can be analyzed in the s domain. Under this approximation, a capacitor with its associated switches in a CAB can be programmed to be a capacitor, positive resistor, or negative resistor. Configurations for realizing these three types of components are described in Fig. 2. When a resistor (either positive or negative) is configured, the absolute value of the resistance is given by

$$R = \frac{T}{C} \tag{1}$$

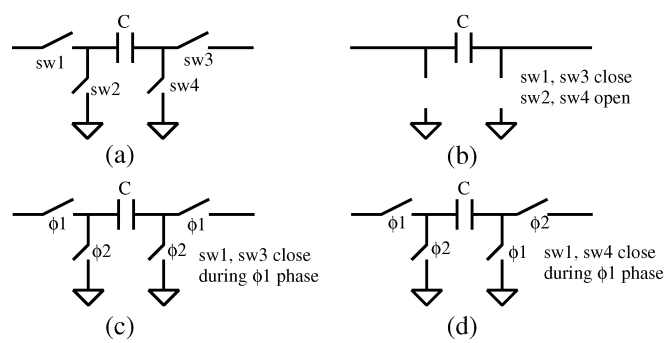


Fig. 2. Programming hardware resources in CABs. (a) Hardware resource. (b) Capacitor configuration. (c) Positive resistor configuration. (d) Negative resistor configuration.

where T is the period of the clock in the FPAA circuit and C is the capacitor value. In this paper, we often treat circuits implemented on SC-based FPAAs as continuous-time circuits. Also, to simplify circuit diagrams, we use conventional resistor and capacitor symbols to represent circuits shown in Fig. 2(b)–(d) whenever detail circuit implementations are not needed in analysis.

III. Proposed Online Testing Approach

In the proposed testing approach, the FPAA circuit under test is first partitioned into subcircuits. Then, unused hardware resources on the FPAA chip are configured to sequentially test partitioned subcircuits. As illustrated in Fig. 3, an FPAA circuit is partitioned into three blocks: *gain stage*, *filter 1*, and *filter 2*. To test the gain stage, spare FPAA resources are configured to implement a replication of the gain stage and a comparator. The replication circuit has the same input as the original circuit. If there are no faults in the circuit, the original circuit and its replication should have the same output (or the difference between the two outputs should be within a certain tolerance range, referred to as a guardband). However, if there are faulty components in either the original or the replication circuit, the outputs of the two circuits will potentially exhibit a large difference and

Authorized licensed use limited to: Southern Illinois University Carbondale. Downloaded on May 28, 2009 at 17:36 from IEEE Xplore. Restrictions apply.

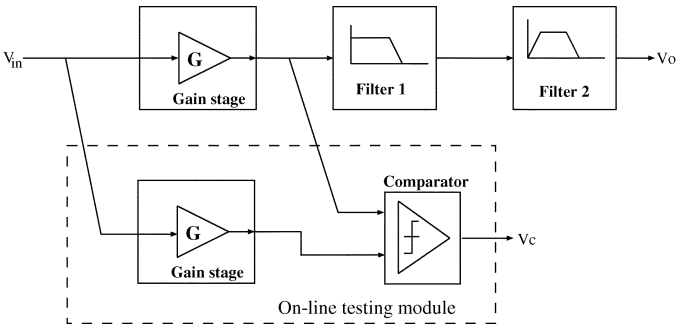


Fig. 3. Online testing scheme for FPAA circuits.

consequently trigger the output of the comparator to indicate the occurrence of faults.

In the implementation of the proposed testing method, a time slot is assigned for testing each partitioned subcircuit. If there are no faults detected after the allocated period for a subcircuit expires, spare FPAA resources will be reconfigured to test another subcircuit. Depending on the vulnerability of subcircuits, time slots with different lengths can be assigned to different subcircuits in the testing process. subcircuits which are prone to faults can be given longer testing periods than those that are robust. Also, according to power budget, the above testing process can be either performed once in a while or conducted continuously in a periodic manner. With the help of simple digital hardware, sophisticated testing schedules can be easily implemented. Also, testing schedules can be altered in the field in order to meet changing reliability requirements.

In the testing process, if the configured testing circuit is faulty, the comparator output will also indicate the occurrence of faults. To eliminate such false alarms, the configured BIST circuit must be guaranteed fault-free. This can be achieved by periodically configuring hardware resources, which are reserved for implementing BIST circuits, to test themselves. In such self-checking operations, online testing for the CUT is temporarily suspended. Hence, the testing for resources reserved for BIST circuits can be carried out in an offline manner. Various techniques developed for analog offline BIST can be exploited for this purpose. Another issue in the implementation of the proposed testing approach is to select proper guardbands. Due to circuit mismatches and other parasitic effects, the CUT output may slightly differ from its replication output in fault-free scenarios. Such differences have been investigated with using statistical analysis and methods to optimally place guardbands have been reported [24], [25].

The proposed method does not intend to utilize powerful processors to synthesize testing circuits in the field. Instead, testing circuits are designed before the implementation of the system. During the operation, configuration bitstreams, which are stored in a memory, are loaded into the FPAA chip to implement various FPAA testing circuits. This approach requires FPAAs having the capability to perform partial dynamic reconfiguration. Currently, some FPAAs already support this feature . With the advance of technology, more sophisticated partially configurable FPAAs will be developed for self-repairing applications. Finally, circuit initial conditions caused by partial reconfiguration need to be carefully treated. For SC-based FPAAs, if op-amps used in the FPAAs have large slew rates and FPAA capacitors have small values, the impact to the CUT caused by adding testing circuits is negligible. This is because such FPAA circuits operate in *sample-and-hold* mode and no additional feedback is introduced when connecting testing circuits to the CUT. Contrary to the FPAA circuit under test, testing circuits do need certain settling time before they can produce correct testing results. To address this problem, a digital timing circuit (e.g., a counter) can be used to temporarily invalidate the testing result during the settling time period of the testing circuit.

IV. Circuit Partition Method

As discussed early, a key step in the proposed testing methodology is to partition the CUT into subcircuits. This paper proposes to partition the FPAA circuit under test into subcircuits such that each subcircuit takes only a single CAB. For example, Fig. 4 shows a third-order low-pass Chebyshev filter [26]. According to its implementation, the filter circuit is partitioned into four subcircuits.

The most apparent advantage of this partition method is its easy implementation. Once analog circuits are mapped onto FPAA chips, circuit partitions are automatically generated. Additionally, this partition approach results in small BIST circuits. Since each subcircuit takes a single CAB, only one spare CAB is needed to replicate the subcircuit under test. Also, the simplest comparator can be implemented using a single CAB. Therefore, BIST circuits resulted from this partition method require only two CABs. Finally, this FPAA partition method also facilitates locating faulty circuit blocks at CAB level. In the testing process, abnormal comparator outputs will automatically identify faulty CABs.

An important metric in evaluating a circuit partition method for testing purposes is how circuit testability is affected after circuit partition. For the proposed testing approach, circuit testability can be measured by $|(V_o - V_o')/V_o|$, where V_o and V_o' are signals at the testing point in fault-free and faulty scenarios. It is shown in [27] that partitioning linear active circuits normally improves circuit testability. This can be justified as follows. Most linear active circuits contain negative feedback paths in order to make circuits stable and insensitive to parasitic effects. However, from the testing point of view, negative feedback generally makes it difficult to detect circuit faults. Performing circuit partitions at CAB level frequently breaks negative feedback paths and consequently enhances circuit testability.

By performing circuit partition at CAB level, subcircuits that are moderately off their performance specifications may be identified as faulty circuits. However, due to either negative feedback or fault-cancellation effects, these subcircuits may not significantly affect the overall performance of the CUT. It can be argued that such partitioned subcircuits should be classified as fault-free circuits because the overall performance of the CUT is not significantly affected. If this is the designer's intention, the threshold of the comparator in the BIST circuit can be accordingly adjusted in order to label these subcircuits

Authorized licensed use limited to: Southern Illinois University Carbondale. Downloaded on May 28, 2009 at 17:36 from IEEE Xplore. Restrictions apply.

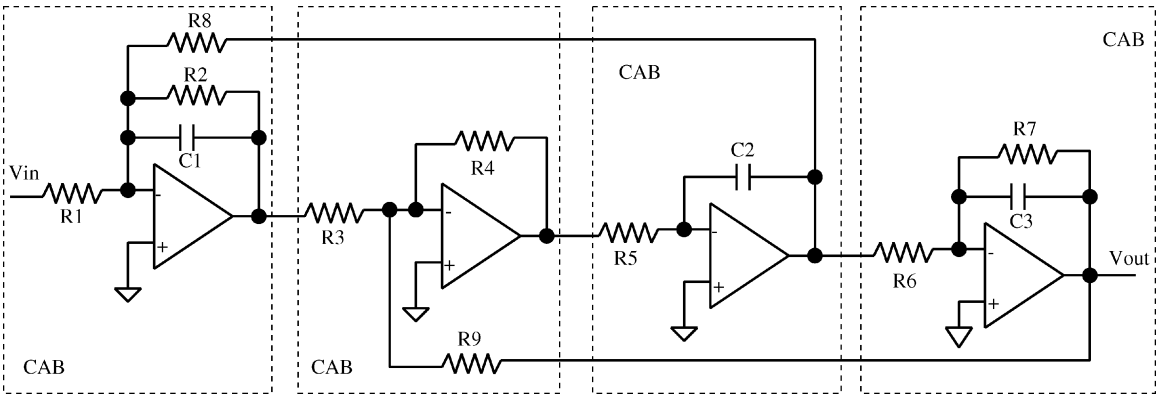


Fig. 4. FPAA circuit partition.

as fault-free circuits. However, the effectiveness of negative feedback and fault cancellation often varies with different input signal characteristics. For systems that require extremely high reliability, a more prudent approach is to treat these subcircuits, which are moderately off their performance specifications, as faulty circuits.

V. Techniques to Address Circuit Stability

After circuit partition, resultant subcircuits may possess unstable transfer functions that have poles in the right-hand-side or the $j\omega$-axis of the s plane. Thanks to negative feedback paths in the unpartitioned FPAA circuit, these subcircuits are stable in the context of the entire FPAA circuit. However, when these subcircuits are duplicated in the testing module, their associated global feedback paths, which cross multiple CABs, are not replicated. As a result, any disturbance or small mismatches between the original circuit and its replication will cause the output of the replication circuit to oscillate or be saturated, consequently invalidating the testing result. A similar problem has been addressed in literature [10] for online testing SC ladder filters. Although the technique presented in [10] can be applied in this testing approach, it normally results in large online testing modules. This not only increases implementation cost but also makes it difficult to find proper placement and routing solutions for the resulted testing circuits. To avoid the above drawbacks, this paper presents a more cost-effective alternative for addressing circuit stability problems.

The proposed circuit technique is explained using the example circuit shown in Fig. 5. The subcircuit under test (circuit in the top rectangle) has a pole located at the origin of the s-plane. Instead of simply replicating the original circuit, we change the amplifier feedback component from a capacitor (C_1 in the original circuit) to a resistive component R_1 in the replication circuit. In SC-based FPAA circuits, R_1 can be implemented as shown in Fig. 2(c). In this implementation, the amplifier feedback path in the replication circuit is broken during ϕ_2 clock phase [refer to Fig. 2(c)]. To address this problem, a small capacitor C_a is added into the amplifier feedback path. The value of C_a is selected small enough such that its impact on the circuit transfer function is negligible. Since the subcircuit under test and its counterpart in the testing module are not exactly the same, conventional comparator circuits cannot be used to compare their outputs. In the proposed comparator circuit, voltage-mode inputs are first converted into current-mode signals. Such conversion is accomplished by using the same type of components that are used in amplifier feedback in the original circuit and its counterpart in the testing module. For example, in Fig. 5, capacitor C_1 is used in the amplifier feedback path of the original circuit. In the comparator circuit, capacitor C_2 converts the output of the original circuit into current signal I_1, whose value is given by

$$I_1(s) = \frac{C_2}{C_1} \cdot \left(\frac{V_{\text{in1}}(s)}{Z_1(s)} + \frac{V_{\text{in2}}(s)}{Z_2(s)} \right). \tag{2}$$

If $C_1 = C_2$, the above equation can be simplified as

$$I_1(s) = \frac{V_{\text{in1}}(s)}{Z_1(s)} + \frac{V_{\text{in2}}(s)}{Z_2(s)}. \tag{3}$$

Similarly, negative resistor R_2, whose implementation is shown in Fig. 2(d), converts the output of the replication circuit into current signal I_2. If $|R_1| = |R_2|$, then we have

$$I_2(s) = -\frac{V_{\text{in1}}(s)}{Z_1(s)} - \frac{V_{\text{in2}}(s)}{Z_2(s)}. \tag{4}$$

The comparator output V_ϵ can be expressed as

$$V_\epsilon(s) = (I_1(s) + I_2(s)) \cdot R_3. \tag{5}$$

Note that capacitor C_b in the comparator circuit is another small capacitor for stabilizing the output of the amplifier when the resistive feedback path is broken. Its effect is neglected in the above analysis. If no faults occur in the circuit, I_1 should equal $-I_2$. Therefore, the comparator output should be zero. On the contrary, if there are faults, I_1 will differ from $-I_2$. The comparator circuit will amplify the current difference and its output will indicate the occurrence of faults.

There are two factors that mainly affect the accuracy of the testing circuit depicted in Fig. 5. The first error source is the amplifier stabilization capacitor C_a, which slightly changes the impedance of the amplifier feedback path in the replication circuit. The second cause for the inaccuracy is the negative resistor used in the comparator circuit. Detailed circuit analysis in [28] shows that the negative resistor causes the comparator circuit

Authorized licensed use limited to: Southern Illinois University Carbondale. Downloaded on May 28, 2009 at 17:36 from IEEE Xplore. Restrictions apply.

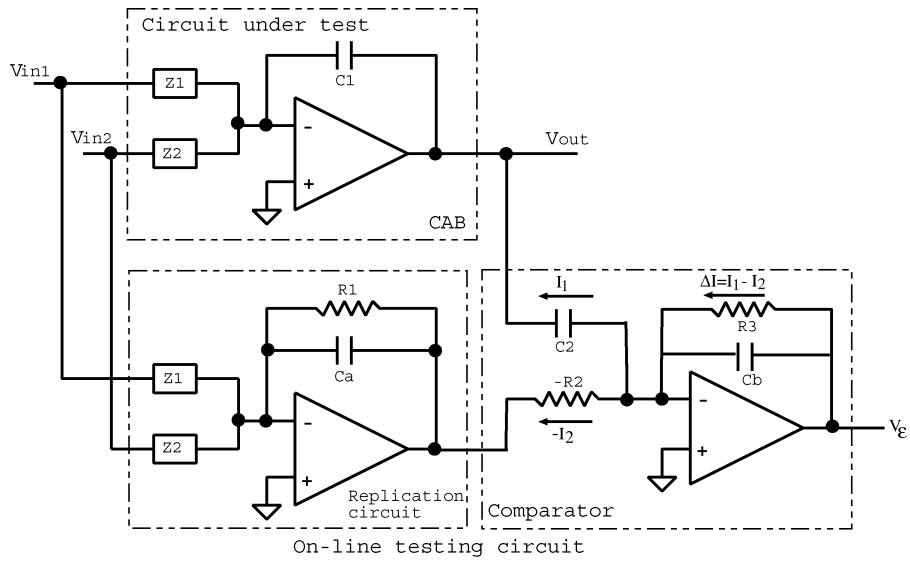


Fig. 5. FPAA online testing circuit.

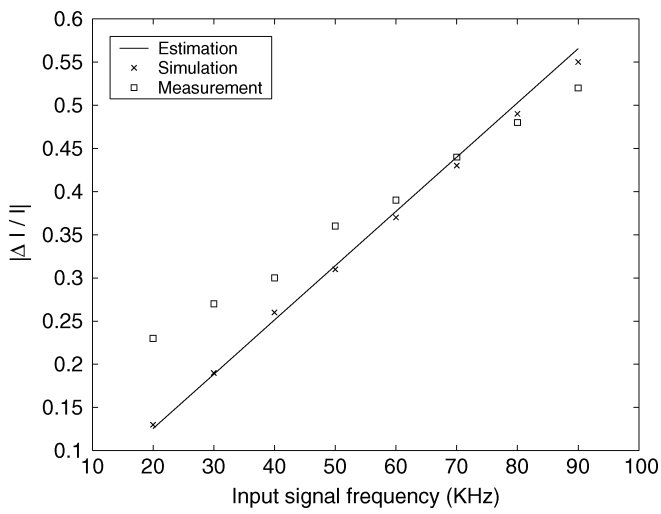


Fig. 6. Comparison of estimation, simulation, and measurement results.

comparing signals generated in different FPAA clock cycles. Due to these error sources, input currents $I_1(s)$ and $I_2(s)$ of the comparator in Fig. 5 are not exactly the same even if the circuit is fault-free. The current mismatch caused by the above two factors is defined as the *intrinsic error* ε of the testing circuit shown in Fig. 5. The intrinsic error, measured by $|(I_1(s) - I_2(s))/I_1(s)|$, can be estimated using the following equation:

$$\varepsilon = 2\pi \frac{f_{\text{in}}}{f_{\text{clk}}} + 2\pi \frac{f_{\text{in}}}{f_{\text{clk}}} \frac{C_a}{C_R} \tag{6}$$

where f_{in} and f_{clk} are the frequencies of the circuit input and FPAA clock; C_a is the amplifier stabilization capacitor and C_R is the FPAA capacitor used to implement resistive component R_1 in Fig. 5. The derivation of the above equation is sketched in Appendix I. The accuracy of the estimation equation is verified by both simulation and measurement results as shown in Fig. 6. Note that there is relatively large difference between measurement and estimation results at the low-frequency range, which is mainly due to measurement errors as discussed in [28].

In (6), term $2\pi(f_{\text{in}}/f_{\text{clk}})(C_a/C_R)$ represents the intrinsic error caused by amplifier stabilizing capacitor C_a; term $2\pi(f_{\text{in}}/f_{\text{clk}})$ is due to the use of the negative resistor in the comparator circuit. If the FPAA circuit under test satisfies constraints of $f_{\text{in}} \leq (1/10)f_{\text{clk}}$ and $C_a \leq (1/10)C_R$, the upper bound of the intrinsic error caused by C_a is approximately 6%. However, when FPAA input signal frequency f_{in} is one-tenth of the FPAA clock frequency, the intrinsic error caused by the negative resistor can be as high as 63%. Since the intrinsic error caused by the negative resistor has a predominant effect on the accuracy of the proposed testing circuit, it is worth taking a close look at the implementation of the negative resistor in the comparator circuit. Fig. 7 shows two comparator circuits with different negative resistor implementations. For the convenience of description, the two comparators are referred to as Comparator 1 and Comparator 2 as shown in Fig. 7. Comparator 1 is the circuit that is used in Fig. 5 and is referred to in the previous discussion. It compares signals generated in different clock cycles and potentially causes large mismatches when signal frequency is high. If the clock schemes of the switches used in the negative resistor circuit are configured as shown in Fig. 7(b), the drawback of comparing signals generated in different clock cycles can be avoided. However, this configuration results in large spikes at the output of the comparator, because capacitors C_1 and C_2 transfer charge to the virtual ground node (the node connecting to the negative input of the amplifier) during different clock phases. To eliminate spikes at the output of Comparator 2, a sample-and-hold circuit [23] can be used to sample the comparator output as shown in Fig. 8. The superiority of this new comparator, referred to as Comparator 3, has been demonstrated through simulation. Fig. 9 compares simulated peak-to-peak voltages that are caused by intrinsic errors at the outputs of Comparators 1 and 3. As Comparator 3 avoids the intrinsic error resulted from the negative resistor its output exhibits significantly smaller peak-to-peak voltage. Note that Comparator 3 is still affected by the intrinsic error caused by stabilization capacitor C_a. That is why the output

Authorized licensed use limited to: Southern Illinois University Carbondale. Downloaded on May 28, 2009 at 17:36 from IEEE Xplore. Restrictions apply.

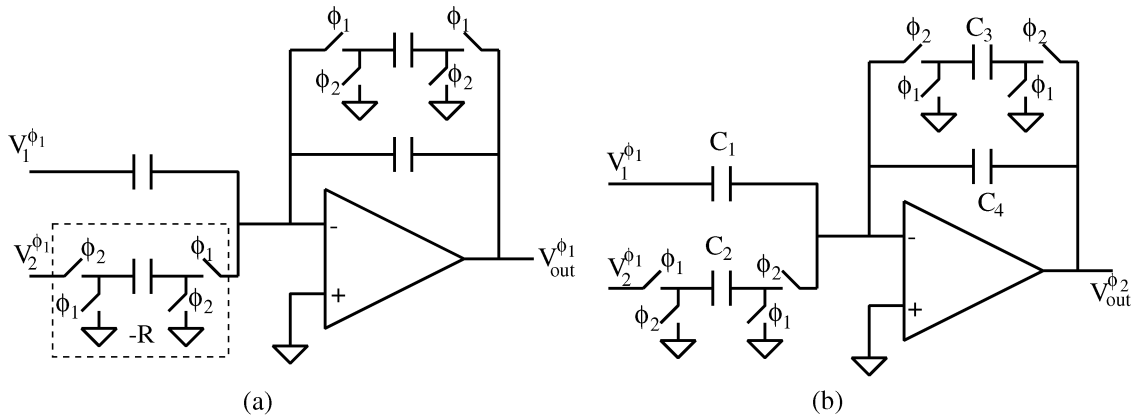


Fig. 7. Comparator implementations. (a) Comparator circuit 1. (b) Comparator circuit 2.

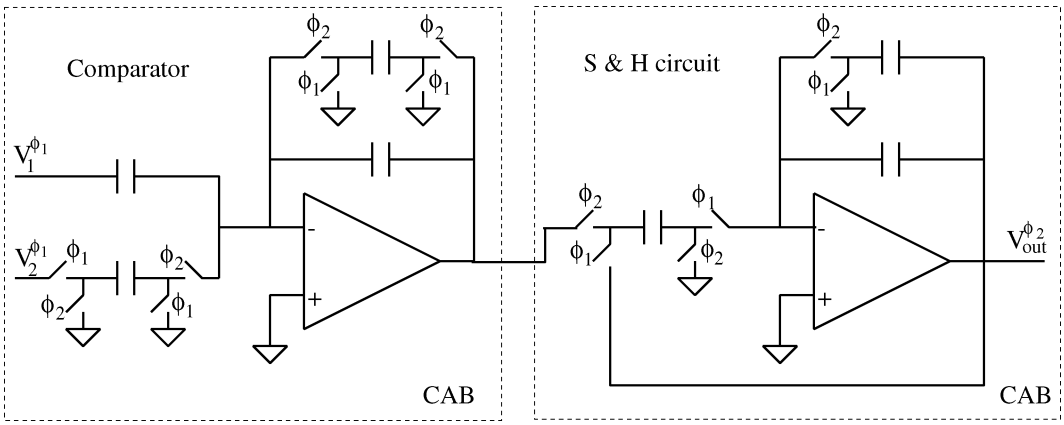


Fig. 8. Comparator 3.

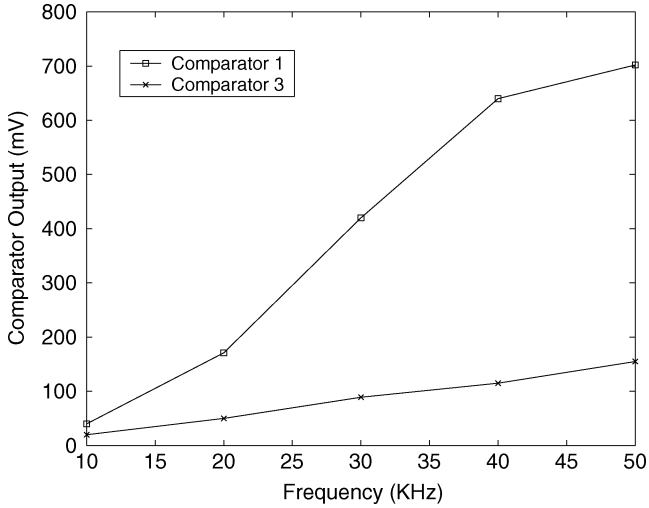


Fig. 9. Simulated comparator outputs.

of Comparator 3 is not exactly zero when it is used to test a fault-free circuit during simulation.

VI. Experimental Results

To demonstrate the feasibility of the proposed testing methodology, experiments are conducted to detect both catastrophic and parametric faults in FPAA circuits. The FPAA device used in our experiments is a Motorola FPAA chip [22], [23].[4] In real applications, the input signal of the CUT can be any type of signals. Since it is impractical to examine all possible input signals, we feed the CUT with sinusoidal, square, triangular, and frequency modulation (FM) signals in the experiment. Results obtained from these scenarios should provide useful inferences for general cases.

One of the circuits that we tested is a low-pass biquad filter, whose implementation is shown in Fig. 10. In the experiment, the corner frequency of the filter is configured to 20 kHz. The filter inputs are sinusoidal, square, and triangular signals with frequency 10 kHz, magnitude 1.5 V, and offset voltage 2.5 V. In order to minimize hardware overhead caused by testing circuits, Comparator 1 is used in our experiments. In fault-free scenarios, the comparator output is close to 2.5 V, which is the signal ground in the FPAA circuit. In faulty scenarios, if the comparator output is beyond the voltage window 2–3 V, the fault is detected. Otherwise, the testing circuit fails to detect the fault.

With the above setup, experiments are first conducted to detect catastrophic faults associated with switches in the filter circuit. In the experiment, selected switches are programmed to be

[4]Since the FPAA used in our experiments is an early product, it does not support partial dynamic reconfiguration. Therefore, each time when we reprogram the testing module, the operation of the entire circuit will be suspended for loading the configuration bitstreams into the chip. This problem will be fully solved by using new FPAA devices that support partial dynamic reconfiguration.

Authorized licensed use limited to: Southern Illinois University Carbondale. Downloaded on May 28, 2009 at 17:36 from IEEE Xplore. Restrictions apply.

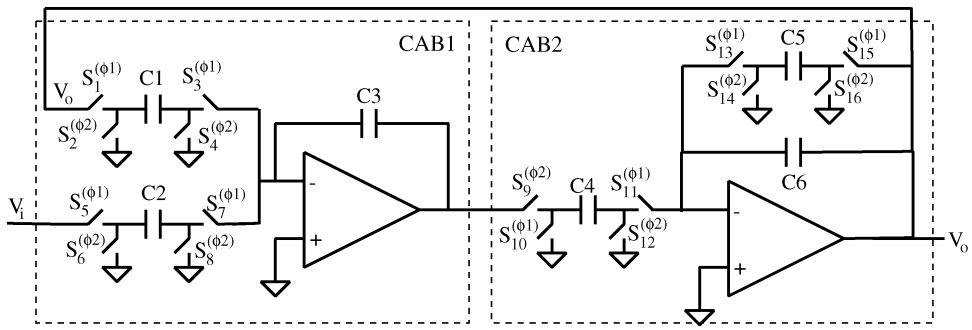


Fig. 10. Low-pass biquad filter.

TABLE I
EXAMINED CATASTROPHIC FAULTS

Fault index	Fault description
1	S_1 stuck-on, S_2 stuck-off
2	S_1 stuck-off, S_2 stuck-off
3	S_3 stuck-on, S_4 stuck-off
4	S_3 stuck-off, S_4 stuck-off
5	S_5 stuck-on, S_6 stuck-off
6	S_5 stuck-off, S_6 stuck-off
7	S_7 stuck-on, S_8 stuck-off
8	S_7 stuck-off, S_8 stuck-off
9	S_9 stuck-on, S_{10} stuck-off
10	S_9 stuck-off, S_{10} stuck-off
11	S_{11} stuck-on, S_{12} stuck-off
12	S_{11} stuck-off, S_{12} stuck-off
13	S_{13} stuck-on, S_{14} stuck-off
14	S_{14} stuck-off, S_{14} stuck-off
15	S_{15} stuck-on, S_{16} stuck-off
16	S_{15} stuck-off, S_{16} stuck-off

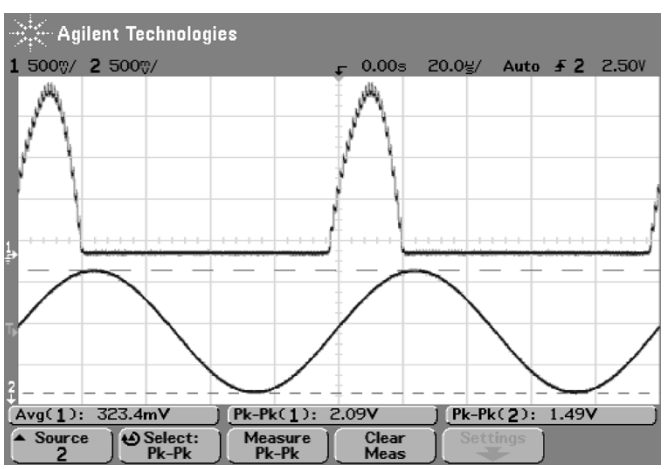


Fig. 11. Captured comparator output.

always-on or always-off to emulate *stuck-on* or *stuck-off* catastrophic faults. Due to programming restrictions on the FPAA device, at least two catastrophic faults have to be injected into the filter circuit in each experiment. Table I lists all catastrophic faults examined in our experiments. All of them are detected by the proposed testing method. Fig. 11 shows captured waveforms when the testing circuit detects fault scenario 16 listed in the table. The top waveform is the comparator output, ranging from 0–2.09 V. The bottom waveform is the filter input, which is a sinusoidal signal with the magnitude of 1.5 V.

TABLE II
MDPFS IN THE LOW-PASS BIQUAD FILTER CIRCUIT

Cap. Names	Cap. values	Input Signal		
		Sinusoidal	Square	triangular
C_1	20	15%	15%	20%
C_2	20	25%	15%	30%
C_3	159	30%	16%	38%
C_4	20	25%	20%	30%
C_5	20	25%	25%	30%
C_6	159	54%	40%	60% †

†Minimum detecable value-decrease fault

TABLE III
MDPFS IN THE LEAPFROG FILTER CIRCUIT

	Equivalent Comp. values	FPAA Cap. values	MDPFs
R_1	$125k\Omega$	8	13%
R_2	$125k\Omega$	8	25%
R_8	$125k\Omega$	8	25%
C_1	$255pf$	255	14%†
R_3	$125k\Omega$	8	13%
R_4	$125k\Omega$	8	13%
R_9	$125k\Omega$	8	13%
R_5	$125k\Omega$	8	13%
C_2	$125pf$	125	14%
R_6	$125k\Omega$	8	13%
R_7	$125k\Omega$	8	25%
C_3	$255pf$	255	9%†

†Minimum detectable value-decrease faults

In the experiment of detecting parametric faults, capacitors in the filter circuit are programmed to incorrect values to imitate capacitor parametric faults. A parametric fault is measured by the variation (in terms of percentage) of the component value. Also, a parametric fault can cause a capacitor value to either increase or decrease. If a fault causes the realized capacitor value to increase, it is referred to as value-increase parametric fault. Similarly, a fault resulting in a decreased capacitor value is called value-decrease parametric fault. For each capacitor in the filter circuit, experiments are conducted to find the minimum detectable value-increase fault as well as the minimum detectable value-decrease fault. The larger one of the two minimum detectable faults is defined as the minimum detectable parametric fault (MDPF) of the given component. Table II summarizes the recorded MDPFs for all the capacitors in the filter circuit. The first and second columns of the table list capacitor names and their normal values, which are represented in terms of FPAA capacitance units. Columns 3, 4, and 5 give the corresponding MDPFs with sinusoidal, square, and triangular input signals.

Authorized licensed use limited to: Southern Illinois University Carbondale. Downloaded on May 28, 2009 at 17:36 from IEEE Xplore. Restrictions apply.

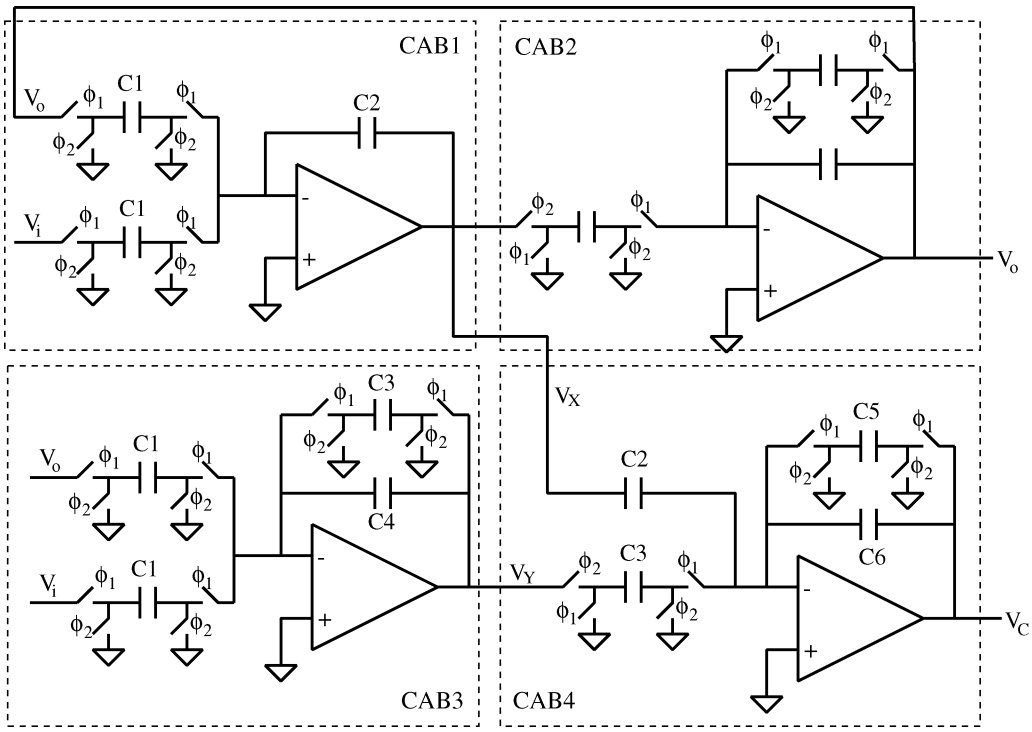


Fig. 12. Online testing low-pass biquad filter.

In addition to the biquad filter, we also successfully tested other types of circuits. Table III lists our experimental results on testing the leapfrog filter circuit, which is shown in Fig. 4. In the experiment, the filter is configured to have a passband of 10 kHz. The filter input during our testing operation was an FM signal, whose carry signal is a sinusoidal signal with magnitude 1.5 V. The modulation signal has a triangular waveform. The carrier frequency is 7 kHz and the maximum frequency deviation due to the modulation is 6 kHz. These parameters are selected to generate a very irregular waveform.

Experimental results show that the proposed testing method can easily detect catastrophic faults. Also, it detects capacitor parametric faults, which mainly range from 13%–30%. For some parametric faults that are difficult to test, the corresponding MDPFs can reach the neighborhood of 60%. Note that the use of Comparator 1 in the testing circuit is a significant cause for the large MDPFs. To combat the intrinsic error caused by the negative resistor, the gain of the comparator has to be relatively small. This makes the comparator less sensitive to mismatches between the subcircuit under test and its replication. If Comparator 3 is used in the testing circuit, the intrinsic error caused by the negative resistor can be eliminated. Potentially, this will improve the MDPFs. Although example circuits presented here are relatively small, the proposed testing method should work with large circuits since we always partition CUTs into subcircuits that occupy single CABs. The size of CUTs should not affect the validity of the testing method.

VII. Concluding Remarks

In this paper, we propose a methodology to perform online self-testing for FPAA circuits. To effectively implement the proposed testing approach, we describe a simple FPAA partition method and present techniques to address circuit stability problems. In addition, error sources in the proposed testing circuits are studied and solutions to improve the accuracy of testing results are presented. The feasibility of the proposed methodology is demonstrated in hardware experiments. This paper mainly focuses on linear active circuits. In the future, we will extend this methodology to testing nonlinear FPAA circuits. Also, efforts will be directed toward improving the effectiveness of this methodology.

The proposed testing method is suitable for the design of self-repairing FPAA circuits. It requires moderate hardware overhead and is capable of locating faulty CABs on the fly. In the implementation of self-repairing FPAA circuits, identified faulty CABs can be replaced by spare resources on FPAA chips. The proposed testing methodology, integrating with advanced FPAA technologies and sophisticated CAB replacement algorithms, will make self-repairing analog circuits more accessible for various applications.

Appendix I

This appendix derives an equation to estimate the intrinsic error in the proposed testing circuit. To facilitate the analysis, it is assumed that the circuit technique depicted in Fig. 5 is used to test a low-pass biquad filter circuit. As shown in Fig. 12, the filter circuit is implemented on CAB1 and CAB2. CAB3 implements the replication of the subcircuit realized on CAB1. The comparator is implemented on CAB4. Using charge conservation principle and z transformation, the expressions for voltage signal $V_X(z)$, $V_Y(z)$, and $V_C(z)$ can be derived as follows:

$$V_X(z) = -\frac{1}{1-z^{-1}}\frac{C_1}{C_2}\left[V_o(z)+V_i(z)\right] \tag{7}$$

Authorized licensed use limited to: Southern Illinois University Carbondale. Downloaded on May 28, 2009 at 17:36 from IEEE Xplore. Restrictions apply.

$$V_Y(z) = -\frac{1}{1-\frac{C_4}{C_3+C_4}z^{-1}}\frac{C_1}{C_3+C_4}\left[V_o(z)+V_i(z)\right] \quad (8)$$

$$V_C(z) = A\cdot\left[C_3\cdot z^{-1}\cdot V_Y(z) - C_2\cdot(1-z^{-1})\cdot V_X(z)\right] \quad (9)$$

where

$$A = \frac{\frac{1}{C_5+C_6}}{1-\frac{C_6}{C_5+C_6}z^{-1}}. \quad (10)$$

Substituting (7) and (8) into (9), we have

$$V_C(z) = A\cdot C_1\cdot B\cdot\left[V_o(z)+V_i(z)\right] \quad (11)$$

where

$$B = \frac{\frac{C_3}{C_3+C_4}z^{-1}}{1-\frac{C_4}{C_3+C_4}z^{-1}} - 1. \quad (12)$$

If there are no faults in the circuit, it is easy to show that B has the same expression as the intrinsic error. Thus

$$\begin{aligned}\varepsilon &= \left|\frac{\frac{C_3}{C_3+C_4}z^{-1}}{1-\frac{C_4}{C_3+C_4}z^{-1}} - 1\right| \\ &= \left|\frac{1-z^{-1}}{1-\frac{C_4}{C_3+C_4}z^{-1}}\right|. \end{aligned} \quad (13)$$

To find the relation between ε and input signal frequency f_{in}, substitute $z^{-1} = e^{-j2\pi(f_{\text{in}}/f_{\text{clk}})}$ into the above equation.

$$\begin{aligned}\varepsilon &= \left|\frac{1-e^{-j2\pi\frac{f_{\text{in}}}{f_{\text{clk}}}}}{1-\frac{C_4}{C_3+C_4}e^{j2\pi\frac{f_{\text{in}}}{f_{\text{clk}}}}}\right| \\ &= \left|e^{-j\pi\frac{f_{\text{in}}}{f_{\text{clk}}}}\right|\cdot\left|\frac{e^{j\pi\frac{f_{\text{in}}}{f_{\text{clk}}}} - e^{-j\pi\frac{f_{\text{in}}}{f_{\text{clk}}}}}{1-\frac{C_4}{C_3+C_4}e^{-j2\pi\frac{f_{\text{in}}}{f_{\text{clk}}}}}\right| \\ &= \left|\frac{j\cdot 2\sin\pi\frac{f_{\text{in}}}{f_{\text{clk}}}}{1-\frac{C_4}{C_3+C_4}e^{-j2\pi\frac{f_{\text{in}}}{f_{\text{clk}}}}}\right|. \end{aligned} \quad (14)$$

If $f_{\text{in}} \ll f_{\text{clk}}$, we use the following approximations:

$$\sin\left(\pi\frac{f_{\text{in}}}{f_{\text{clk}}}\right) \approx \pi\frac{f_{\text{in}}}{f_{\text{clk}}} \quad (15)$$

$$e^{-j2\pi\frac{f_{\text{in}}}{f_{\text{clk}}}} \approx 1. \quad (16)$$

Hence, (14) can be simplified as

$$\varepsilon = 2\pi\frac{f_{\text{in}}}{f_{\text{clk}}} + 2\pi\frac{f_{\text{in}}}{f_{\text{clk}}}\frac{C_4}{C_3}. \quad (17)$$

Note that C_3 and C_4 in (17) are the components denoted as C_R and C_a in the previous discussion.

REFERENCES

[1] D. D'mello and G. Gulak, "Design approaches to field-programmable analog integrated circuits," *Analog Integr. Circuits Signal Process.*, vol. 17, no. 2, pp. 7–34, 1998.

[2] B. Pankiewicz, M. Wojcikowski, S. Szczepanski, and Y. Sun, "A field programmable analog array for CMOS continuous-time OTA-C filter applications," *IEEE J. Solid-State Circuits*, vol. 37, no. 2, pp. 125–136, Feb. 2002.

[3] M. Mar, B. Sullam, and E. Blom, "An architecture for a configurable mixed-signal device," *IEEE J. Solid-State Circuits*, vol. 38, no. 3, pp. 565–568, Mar. 2003.

[4] R. S. Zebulum and D. Keymeulen *et al.*, "Experimental results in evolutionary fault-recovery for field programmable analog devices," in *Proc. NASA/DoD Conf. Evolvable Hardware*, 2003, pp. 182–186.

[5] T. Balen, A. Andrade, F. Azais, M. Lubaszewski, and M. Renovell, "An approach to the built-in self-test of field programmable analog arrays," in *Proc. VLSI Test Symp.*, 2004, pp. 383–388.

[6] T. Slaughter and C. Stroud, "Fault injection emulation for field programmable analog arrays," in *Proc. Southwest Symp. Mixed-Signal Design*, 2003, pp. 212–216.

[7] R. Sanahuja, V. Barcons, L. Balado, and J. Figueras, "X-Y zoning BIST: an FPAA experiment," in *Proc. Int. Mixed-Signal Test Workshop*, 2002, pp. 237–243.

[8] A. Andrade and G. Viera *et al.*, "Testing global interconnects of filed programmable analog arrays," in *Proc. 10th Int. Mixed-Signal Testing Workshop*, 2004, pp. 231–236.

[9] J. L. Huertas, A. Rueda, and D. Vazquez, "Testable switched-capacitor filters," *J. Solid-State Circuits*, vol. 28, no. 7, pp. 719–724, 1993.

[10] D. Vazquez, A. Rueda, and J. L. Huertas, "A solution for the on-line test of analog ladder filters," in *Proc. VLSI Test Symp.*, 1995, pp. 48–53.

[11] D. Vazquez, A. Rueda, J. L. Huertas, and E. Peralias, "A high-Q bandpass fully differential SC filter with enhanced testability," *J. Solid-State Circuits*, vol. 33, no. 7, pp. 976–986, 1998.

[12] J. Velasco-Medina, I. Rayane, and M. Nicolaidis, "On-line BIST for testing analog circuits," in *Proc. Int. Conf. Comput. Design*, 1999, pp. 330–332.

[13] M. Negreiros, L. Carro, and A. A. Susin, "A statistical sampler for a new on-line analog test method," *J. Electron. Testing*, vol. 19, pp. 585–595, 2003.

[14] B. Vinnakota and R. Harjani, "The design of analog self-checking circuits," in *Proc. Int. Conf. VLSI Design*, 1994, pp. 67–70.

[15] O. K. Abu-Shahla and I. M. Bell, "An on-line self-testing switched-current integrator," in *Proc. IEEE Int. Test Conf.*, 1997, pp. 463–470.

[16] M. Lubaszewski, S. Mir, V. Kolarik, C. Nielsen, and B. Courtois, "Design of self-checking fully differential circuits and boards," *IEEE Trans. Very Large Scale Integr. Syst. (VLSI)*, vol. 8, no. 2, pp. 113–128, Apr. 2000.

[17] A. Chatterjee, "Checksum-based concurrent error detection in linear analog systems with second and higher order stages," in *Proc. VLSI Test Symp.*, 1992, pp. 286–291.

[18] H. Yoon, A. Chatterjee, and J. L. A. Hughes, "Optimal design of checksum-based checkers for fault detection in linear analog circuits," in *Proc. Inte. Conf. VLSI Design*, 1997, pp. 393–397.

[19] J. Velasco-Medina, M. Nicolaidis, and M. Lubaszewski, "An approach to the online testing of operational amplifiers," in *Proc. 7th Asian Test Symp.*, 1998, pp. 290–295.

[20] E. F. Cota, M. Negreiros, L. Carro, and M. Lubaszewski, "A new adaptive analog test and diagnosis system," *IEEE Trans. Instrum. Meas.*, vol. 49, no. 2, pp. 223–227, Apr. 2000.

[21] S. Krishnan, S. Sahli, and C. L. Wey, "Test generation and concurrent error detection in current-mode A/D converters," in *Proc. Int. Test Conf.*.

[22] A. Bratt and I. Macbeth, "DPAD2—a field programmable analog array," *Analog Integr. Circuits Signal Process.*, vol. 17, no. 2, pp. 67–89, 1998.

[23] *EasyAnalog Design Software User's Manual*, Motolora Inc., Schaumberg, IL, 1997.

[24] P. Mullenix, "The capability of capability indics with an application to guardbanding in a test environment," in *Proc. Int. Test Conf.*, 1990, pp. 907–915.

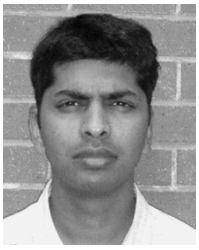
[25] R. H. Williams and C. F. Hawkins, "The economics of guardband placement," in *Proc. Int. Test Conf.*, 1993, pp. 218–225.

[26] S. Natarajan, *Theory and Design of Linear Active Networks*, New York: Macmillan, 1987.

[27] H. Wang, S. Kulkarni, and S. Tragoudas, "On-line testing field programmable analog array circuits," in *Proc. Int. Test Conf.*, 2004, pp. 1340–1348.

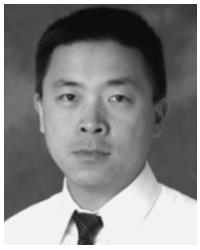
[28] ——, "Circuit techniques for field programmable analog array on-line testing," in *Proc. 10th Int. Mixed-Signal Testing Workshop*, 2004, pp. 237–244.

Authorized licensed use limited to: Southern Illinois University Carbondale. Downloaded on May 28, 2009 at 17:36 from IEEE Xplore. Restrictions apply.



Amit Laknaur received the B.S. degree in electrical engineering from Chaitanya Bharati Institute of Technology, Hyderabad, India, in 2003. He is currently pursuing the M.S. degree at Southern Illinois University, Carbondale.

His research interests are mixed-signal circuit design and testing.



Haibo Wang received the B.Eng. degree from Tsinghua University, Beijing, China, in 1992, the M.Eng. degree from Nanyang Technological University, Singapore, in 1997, and the Ph.D. degree from the University of Arizona, Tucson, in 2002.

Currently, he is an Assistant Professor with the Department of Electrical and Computer Engineering at Southern Illinois University, Carbondale. His research interests are in design and testing of mixed-signal circuits.

Dr. Wang received NSF CAREER award in 2005.

Authorized licensed use limited to: Southern Illinois University Carbondale. Downloaded on May 28, 2009 at 17:36 from IEEE Xplore. Restrictions apply.