



HAL
open science

Testing the Configurable Analog Blocks of Field Programmable Analog Arrays

Tiago R. Balen, Antonio Andrade Jr., Florence Azaïs, Marcelo Lubaszewski,
Michel Renovell

► **To cite this version:**

Tiago R. Balen, Antonio Andrade Jr., Florence Azaïs, Marcelo Lubaszewski, Michel Renovell. Testing the Configurable Analog Blocks of Field Programmable Analog Arrays. ITC: International Test Conference, Oct 2004, Charlotte, United States. pp.893-902. lirmm-00108897

HAL Id: lirmm-00108897

<https://hal-lirmm.ccsd.cnrs.fr/lirmm-00108897v1>

Submitted on 21 Jun 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

TESTING THE CONFIGURABLE ANALOG BLOCKS OF FIELD PROGRAMMABLE ANALOG ARRAYS

T. Balen¹, A. Andrade Jr.¹, F. Azaïs², M. Lubaszewski^{1,3}, M. Renovell²

¹DELET-UFRGS

Univ. Fed. do Rio Grande do Sul
Porto Alegre, Brazil
{balen, andradejr, luba}@eletro.ufrgs.br,

²LIRMM

Université de Montpellier II
Montpellier Cedex 5, France

³IMSE-CNM

Inst. de Microelectr. de Sevilla
Sevilla, Spain
luba@imse.cnm.es

Abstract

The problem of testing the Configurable Analog Blocks (CABs) of Field Programmable Analog Arrays (FPAAs) is addressed in this paper. The considered fault model comprises deviations in the nominal values of CAB programmable capacitors, deviations in the programmable gains of CAB input amplifiers and stuck-on/stuck-open faults in CAB switches. The problem of test stimuli generation is solved, in a first approach, by using the Oscillation Test Strategy (OTS), which is associated to a test response analysis external to the device under test. In a second approach, a Built-In Self-Test (BIST) scheme is proposed by associating to the OTS an Output Response Analyzer (ORA) built using the internal FPAA resources. Both approaches are validated using the ispPAC10 FPAA from the Lattice Semiconductor Corporation. In the paper, the approaches are compared in terms of fault coverage, test application time and required external hardware resources for testing. Experimental results show that a good compromise of these aspects can be found by taking the best of each approach.

1. Introduction

As the electronics expands in the modern society, the application of analog circuits becomes ever more important. In many electronic systems, such as those used in communications, entertainment, speech and image processing, for example, the analog parts are mainly responsible for the interface between the physical world and the digital circuitry. In high-performance systems, on the other hand, analog circuits can also be used to accelerate the processing time. This evolution has pushed analog circuits to a higher level of complexity and, consequently, to the need of higher accuracy and quality.

In the last years, a new type of analog circuits started becoming popular: the Field Programmable Analog Arrays (FPAAs). In terms of design time, FPAAs provide to the analog world the same advantages as their digital counterparts, the Field Programmable Gate Arrays (FPGAs), provide to digital circuits. They boost design flexibility allowing the fast prototyping of analog circuits and reconfigurability, not only during system design, but also during the application in the field.

A typical FPAA structure comprises Configurable Analog Blocks (CABs), I/O blocks, an interconnection network and memory registers for digital device programming. Figure 1 depicts a typical block diagram for a generic FPAA. The typical architecture of a generic CAB is depicted in Figure 2.

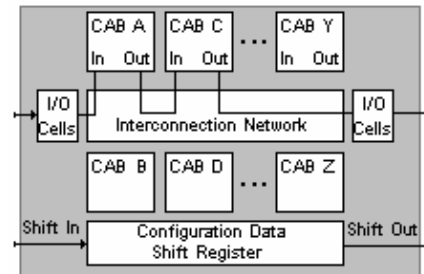


Figure 1: Building blocks of a generic FPAA

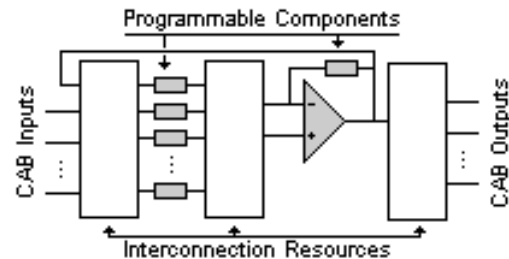


Figure 2: Typical architecture of a generic CAB

Each CAB is composed of an array of programmable analog components, local and global interconnection

switching blocks and associated wiring, and an Operational Amplifier (OpAmp). The components within the array can be implemented as simple wires, passive or active components or other more complex parts. In general, the programmable parameters of CABs are gain of amplifiers, values of resistors and capacitors, as well as setting global and local feedback loops.

As an example of FPAA, the device studied in this work is the ispPAC10 from Lattice Semiconductor Corporation [1]. It has in its structure four CABs, called PACBlocks. These CABs are composed of two Input Amplifiers (IA) with programmable gain, an Output Amplifier (OA), a programmable capacitor in the feedback loop and a feedback resistor. Figure 3 shows the schematic diagram for the Lattice FPAA CAB.

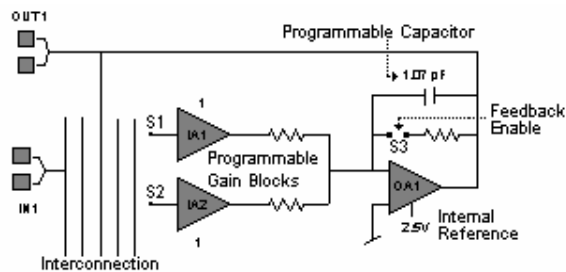


Figure 3: Lattice configurable analog block

As FPAAs become an important platform for analog circuit prototyping, it is crucial to ensure the device fault-free operation. In the last decade, much effort has been expended in the development of efficient and cost-effective test and Built-In Self-Test (BIST) techniques for analog and mixed-signal circuits, including filters, data converters, etc [2, 3]. However, just some few works have recently started facing the problem of testing FPAAs [4, 5]. For testing purposes, the FPAA partitioning proposed in figure 1 also applies, since each individual part requires a specific test technique. This work, however, focuses only on the CAB testing due to its complexity and relevance to the device functionality.

In [4], an oscillation strategy (OTS) is applied to test some structures of an FPAA CAB. In this method, the blocks of the programmable device are configured as oscillators and the frequency and amplitude of the generated signal are taken as the signature of the circuit. The generated signal is a function of the internal components of the CAB, and a fault is detected if this signature deviates from that of the fault-free circuit. In [5], some CABs of the FPAA are configured as an oscillator, and others, as an Output Response Analyzer (ORA). The ORA provides a compact signature that depends on the signal frequency, amplitude and phase. This BIST approach requires zero area overhead, since the test structures are built with the internal blocks of the FPAA.

In the previously mentioned works, only the programmable capacitors of the CABs are tested. This work extends the OTS and ORA techniques to the testing of block gains and programmable switches of the FPAA CABs. Additionally, improvements to the BIST scheme in [5] are made, so that the repeatability of signature measures is ensured. The paper is organized as follows: in section 2, the application of the OTS to FPAAs is presented, whereas in section 3 an enhanced built-in ORA structure is used to observe and compact the signatures for the results obtained with the OTS. Section 4 evaluates the experimental results obtained with the OTS and ORA approaches and compares one against another in terms of fault coverage, testing time and hardware resources required. Finally, section 5 presents the concluding remarks.

2. Oscillation Test Applied to FPAAs

Due to the programmable nature of FPAAs, one can pose that there are countless possible test vectors to be applied to the CABs. An alternative to deal with the CAB testing is a vectorless technique applied to analog circuits, the *Oscillation Test Strategy* (OTS) [6]. In this method, blocks of the circuit under test are individually converted into oscillators. These oscillators generate oscillation frequencies that can be expressed as a function of the important parameters of the blocks. In order to increase the fault coverage or to make the fault detection easier, the amplitude of the oscillations may be taken as a test measurement complementary to the frequency [7].

In [4], this technique was applied to test the programmable capacitors of the CABs of a Lattice FPAA. The oscillator was built with two CABs, using a quadrature topology, composed of two pure integrators connected in a ring configuration (figure 4). The frequency of the generated output is a function of the values of the programmable elements. In the Lattice device, since four CABs are available, one can build two oscillators and evaluate the generated signal of these two oscillators at the same time.

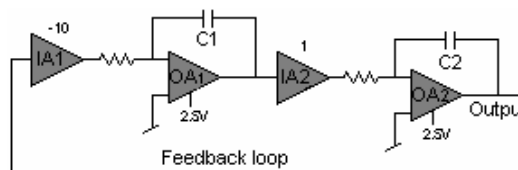


Figure 4: Quadrature oscillator topology

In order to establish the signatures of the fault-free circuit, it is necessary to compute the frequency and amplitude of the generated signal for all existing values of the programmable elements of a CAB. During the test application, if the obtained signals are out of a tolerance window defined around these parametric curves, then it

means that the deviations in the programmed elements cannot be accepted, and the device is classified as faulty.

With this approach, one can implement as many oscillators as needed, in order to perform the test of all configurable blocks. It is also possible to test more than one block at a time, decreasing the test application time. Nevertheless, an external ORA is still necessary.

In order to check the potential defect coverage that this test method can achieve in the Lattice FPAA, fault injection is performed by means of the programming software. The fault injection is based on configuring the device with a fault-free value of component and then reconfiguring again with faulty values greater and lesser. Since details of the physical implementation are not known, the fault model, and thus injection, is limited to the programming possibilities available for the components of the device under test.

2.1. Capacitor Testing

The first step to capacitor testing is to obtain the fault-free signature of the circuit for all possible values of oscillator capacitors. For the sake of simplicity we consider $C_1=C_2$ (figure 4). The programmable capacitors range from 1.07pF to 61.59pF, in a total of 127 discrete values. All possible configurations of the circuit are downloaded to the FPAA and the frequency and amplitude at the output are measured and computed as the fault free behavior.

According to the programming possibilities of the device, two faults are considered for each capacitor: $\pm 20\%$ deviations of the nominal value. As the values of the

programmable capacitors are fixed, in some cases it is not possible to inject exact $\pm 20\%$, thus deviations in the range from 15 to 25% are, in those cases, admitted in the fault model. Outside this range, we consider that the fault cannot be injected and it is neither computed in the fault coverage, nor in the counting of Test Configurations (TCs). For example, from 1.07 to 3.11pF (3 programmable values), neither +20% nor -20% parametric faults can be injected; for two values, 5.06pF and 5.46pF, only +20% deviations are injected; from 53.93 up to 61.59pF (11 values), only -20% faults can be injected. In the remaining range from 5.92 to 53.53pF (111 values), all faults of the model are injected. Thus, a total of 940 faults are injected into the device (2 faults \times 111 programmable values, plus 1 fault \times 13 programmable values, for each one of the 4 CABs), by modifying the nominal value of one capacitor at a time in the programming software. Figures 5 and 6 show, respectively, the measured values of the frequency and the amplitude of the output signal for the fault-free and faulty circuit for each possible value of capacitor C_1 . These results also apply to capacitor C_2 and to the other two capacitors of the second oscillator, built with the remaining FPAA CABs.

Comparing the obtained results, one can conclude that the frequency parameter is suitable to detect faults in capacitors from the lowest to the midrange values of capacitances, while the amplitude parameter is better to detect the remaining faults. Thus, the measure of both output parameters ensures a fault coverage of 100% for the capacitor faults considered. Finally, one will need to configure the device 124 times to test the CAB capacitors.

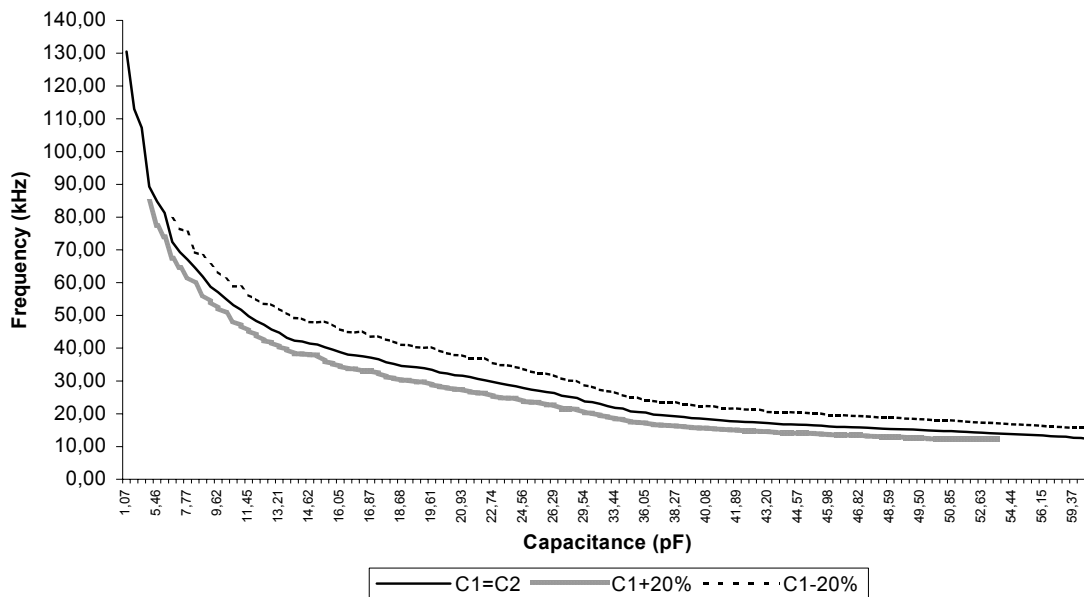


Figure 5: Output signal frequency x C1 values

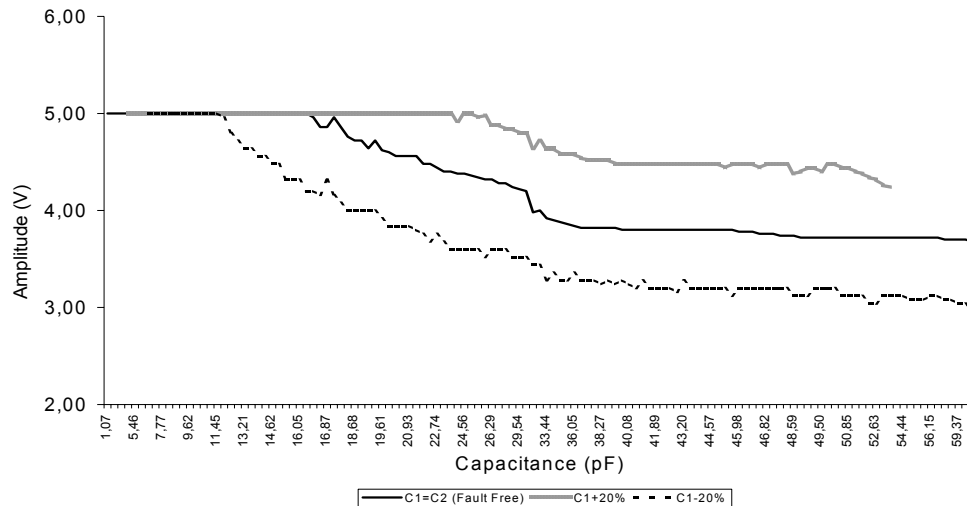


Figure 6: Output signal amplitude x C1 values

2.2. Gain Testing

Testing the CAB gains with OTS presents a particular problem: a low sensitivity of the oscillation frequency is observed for high values of gain and the amplitude of oscillations does not change much from one gain to another. In order to obtain the best sensitivity, a series of experimental tests was performed by changing the capacitor values and observing the signal behavior. One observes that the largest sensitivity of the oscillation frequency to gain variations occurs when the capacitors are set to their minimum value (1.07 pF).

Considering this problem, a curve is plotted to predict the fault-free behavior of the oscillation frequency with respect to gain variations (curve K in figure 7). The signal amplitude is not considered because it does not add to the observability of faulty behaviors. The gain of the first CAB of every oscillator is fixed to -1 , and the gain of the other varies from $+1$ to $+10$. To ensure that the circuit still oscillates, the gain of the blocks must have opposite signals. Hence, to test the gain of the second block from -1 to -10 , one needs to set the gain of the first block to $+1$. It is observed that, for low values of gain magnitude, there is a large sensitivity, while for gain values greater than 5 in modulus, the sensitivity is much lower. Therefore, one can predict that fault detection is more difficult for deviations close to high values of gain in modulus.

Since one is allowed to program just some few discrete values for the gain of the PAC-Blocks, only faults of the type $K_{\text{faulty}} = K \pm 1$ are injected into the oscillator to analyze faulty behaviors. Nevertheless, it should be pointed that, for the device in use, deviations of $+1$ for a gain value of $+10$ (or a deviation of -1 for -10), as well as -1 deviation for a gain of $+1$ (again, $+1$ for -1), cannot be

injected, as the corresponding faulty values are not available for programming. Thus, the number of injected faults per IA is $40-4=36$. As the device comprises 8 IAs, $36 \times 8 = 288$ faults are injected for fault coverage evaluation.

The fault simulation curves are shown in figure 7. As predicted, they show that fault detection for gain values greater than 5 is hard to achieve. Considering a tolerance band of $\Delta f_{\text{min}} = 500$ Hz, only faults for $|K| \leq 5$ are detected. If another oscillator topology is defined, it is possible that faults in the interval $|K| \geq 5$ are detected. This may occasionally require that external components are added to the oscillator.

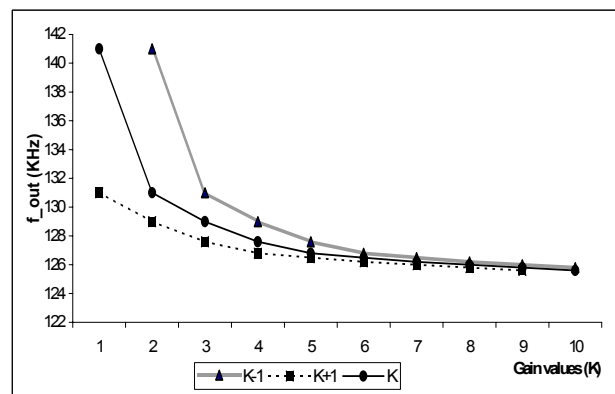


Figure 7: Fault-free and faulty behavior of the oscillation frequency with respect to gain

As there are two CABs in each oscillator and every CAB has two IAs, one has to perform the same test procedure for 4 IAs sequentially. Since the CABs that comprise the second oscillator can be set and tested in parallel, and this approach detects faults for $-5 \leq K \leq 5$, the number of necessary TCs for the gain totals up 40. Since only 144 out of 288 fault can be detected, a fault coverage of 50% is achieved.

2.3. Switch Testing

Testing the CAB switches using OTS becomes very simple, since a switch in the signal path being off causes the circuit not to oscillate. To test switches S_1 and S_2 (figure 3), the oscillator is configured to use one of the two possible IAs. Testing the other switch is achieved by exchanging the oscillator input with the other IA input. Thus, if a stuck-open fault affects one of these two switches, the circuit will not oscillate. To detect stuck-on faults, the same circuit is used, but the switches are kept off, in both configurations. The circuit should not oscillate.

For switch S_3 (local OA feedback, in figure 3), the testing procedure is analogous, but instead of a stuck-open, as it is for S_1 and S_2 , a stuck-on fault determines the absence of oscillation.

Note that only switches internal to the CAB or used for locally connecting it to implement the OTS oscillator are taken into account here. Other switches are considered as part of the FPAA interconnection network and may be tested as it is proposed, for instance, in [8]. Table 1 shows the necessary configurations for testing the internal (S_3) and local (S_1 and S_2) CAB switches. During the test of one CAB in the oscillator, the other CAB remains with the original settings.

Table 1: Configurations for switch testing

Fault	Switches Setup	Fault-free Behavior	Faulty Behavior
<i>St on S1</i>	S1, S2, S3 off	No Osc.	Oscillation
<i>St on S2</i>		No Osc.	Oscillation
<i>St on S3</i>	S1 on, S2, S3 off	Oscillation	No Osc.
<i>St open S1</i>		Oscillation	No Osc.
<i>St open S2</i>		Oscillation	No Osc.
<i>St open S3</i>	S1, S3 off, S2 on	No Osc.	Oscillation

From table 1, one can conclude that 4 different TCs are needed in order to test all the switches of a CAB. As the test is analogous for the other CAB, 4 additional TCs apply. Since the second oscillator implemented in the ispPAC10 can be checked concurrently with the first one, the CAB switches require 8 TCs altogether. It can be seen in table 1 that all switch faults considered are detected.

3. An Approach to the BIST of FPAAs

Research on digital BIST has led in the past to the proposal of multifunctional structures capable of scanning test data, generating test patterns and compacting test responses. These structures were called Built-In Logic Block Observers (BILBOs). An analog version of these structures was proposed in [9], called ABILBO (Analog Built-In Block Observer). This analog test structure has the same features of the digital BILBO. The capability of generating test stimuli and compacting test responses are those of interest for this work. Basically, the ABILBO is

composed of two integrators and some additional circuitry, and can be programmed as a quadrature ring oscillator (similarly to figure 4), or as a double integrator signature analyzer (as in figure 8).

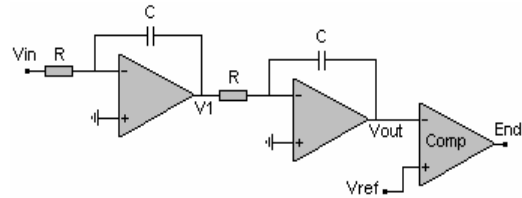


Figure 8: Double integrator signature analyzer

A signature can be obtained by computing the time for the output of the second integrator to reach a predefined V_{ref} . This time depends on the amplitude, the frequency and the phase of the analyzed signal [9].

Assuming a signal $V_{in} = -V_o \sin(\omega t + \phi)$ and $V_C(t=0) = 0$ for the double integrator capacitors, it can be shown that:

$$V_1 = \frac{V_o}{(\omega\tau)} [\cos\phi - \cos(\omega t + \phi)] \quad (1)$$

$$V_{out} = \frac{V_o}{(\omega\tau)^2} [-\sin(\omega t + \phi) + \omega t \cdot \cos\phi + \sin\phi] \quad (2)$$

where V_1 is the signal at the output of the first integrator, V_{out} is the signal at the output of the second one, and $\tau = RC$ is the time constant for the integrators. Figure 9 depicts the effect of the double integration over a signal with $V_o = 1V$ and $f = 750$ Hz. Considering $V_{ref} = 2V$ and $\tau = 0.66ms$, the time signature for V_{in} will be $t = 4.5ms$.

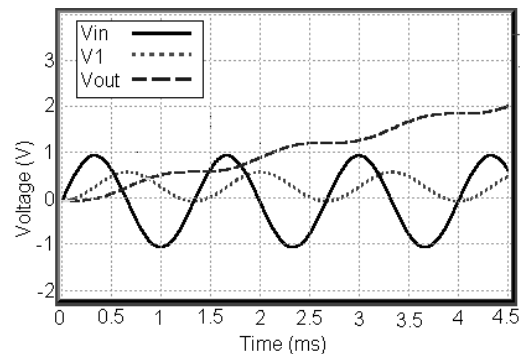


Figure 9: Effect of double integration for a sinusoidal signal

In [9], a detailed evaluation on the fault coverage, the sensitivity for different time constants and the probability of aliasing is given for this ORA structure.

Since the Lattice ispPAC10 has four CABs that can easily implement an integrator, it is possible to build two ABILBO structures in this device (figure 10). For the purpose of signature analysis, two CABs are configured as an oscillator (as in figure 4) and the remaining CABs are programmed as a double integrator ORA (similarly to figure 8). The ORA comparator is implemented outside

the device and V_{ref} is set to 4V. The output of the oscillator is connected at the input of the ORA.

Note that the first ORA stage is now a lossy integrator. In the experiments performed in [5], it was observed that the repeatability of the signature measurements was poor for some ranges of capacitances. During the start-up of the oscillator, the original ORA integrates twice the input signal, which is made up of a combination of the offset of its first integrator and a circuit reference voltage that appears drifted due to noise. The resulting time signature then changes reasonably for the same oscillator, since a parabola component is added to the expected ORA output due to the double integration of that random DC level. Therefore, another topology, where the first integrator of the ORA is now lossy, is chosen so as to reduce this effect, as shown in figure 11. For a given value of capacitor, three time signatures are acquired for the original (figure 11a) and for the new ORA (figure 11b). One can see that when a lossy integrator is used, the drift of the time signature is made small enough to ensure repeatability of measures.

For each different value of the programmable components of the oscillator and the signature analyzer, there is one different time signature. This way, a fault that affects the components of the ABILBO structures shall generate a signature that differs from that previously computed as the fault-free signature. In order to validate the new ORA structure, the same fault model as in the OTS method is used next.

3.1 Signature Analysis for Capacitor Faults

It is noticed that, for values smaller than 30pF, faults on integrator capacitors have poor observability. Thus, for these capacitor values only the two CABs implementing the oscillator are tested at the same time, while the CABs implementing the ORA are kept fixed at the maximum value of τ ($C_3=C_4=62pF$), as larger ORA time constants increase fault detection [9]. Next, one can change the device configuration so that the CABs that are first configured as the ORA are then reconfigured as the oscillator. In this manner, a good fault coverage is ensured with a few number of TCs.

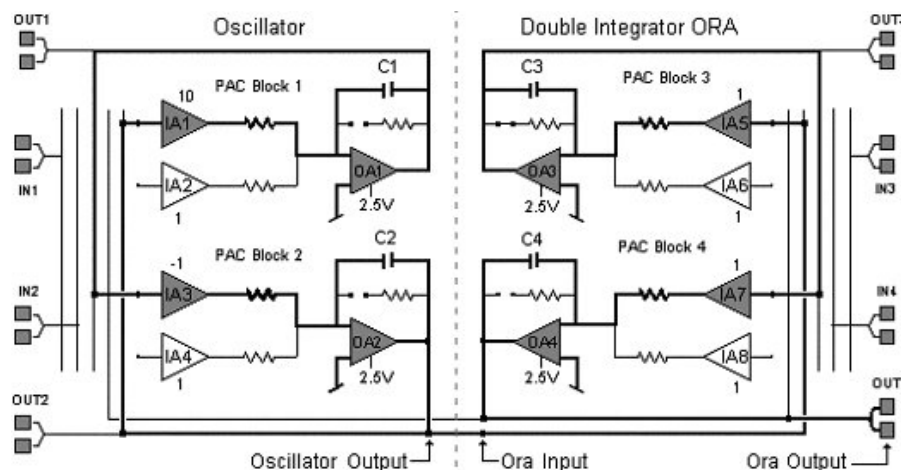


Figure 10: PACBlocks and implementation of oscillator and ORA

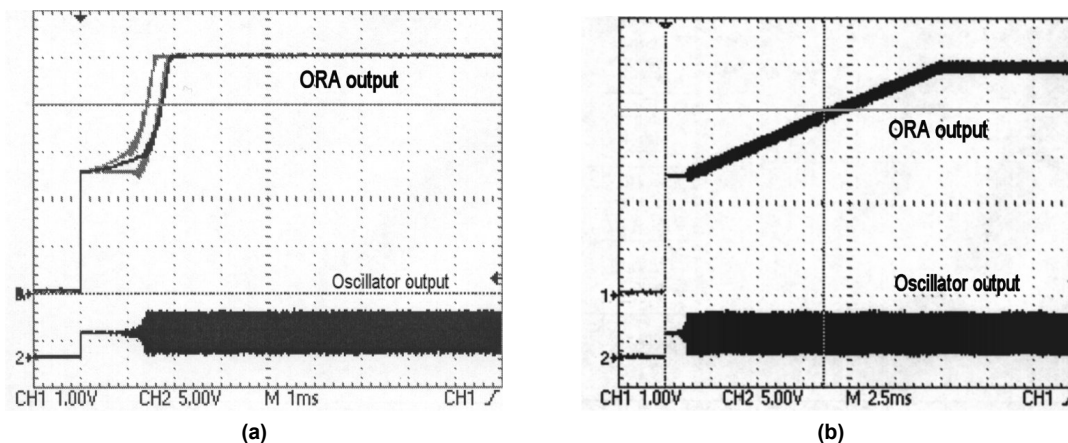


Figure 11: Acquired time signatures for (a) original ORA and (b) ORA with lossy integrator

Additionally, fault coverage can be further improved if another test parameter is considered, the start-up time of oscillations. This parameter, that changes for different values of oscillator capacitors and IA gains, is the time elapsed from the moment the device is programmed until oscillation actually starts. The oscillator start-up time can be observed in figures 11(a) and (b), which show an amplitude settling time of approximately 1.4ms. Therefore, taking advantage of this parameter, for capacitances smaller than 14pF the oscillator closed loop gain is set low enough to make the start-up time observable in the signature and, at the same time, large enough to ensure oscillation. For values larger than 14pF, a large closed loop gain ensures oscillation, as well as it makes the ORA output sensitive to start-up time changes.

Given all the points stated above, three test sets are carried out to obtain the best results possible for the test of capacitors: from 1pF up to 13pF, the oscillator closed loop gain is set to -2 and the ORA has τ fixed at the maximum value; from 14pF up to 30pF, the gain is now set to -10 and τ keeps fixed at maximum; for values above 30pF, the gain is also set to -10 , but τ is changed since faults in the ORA capacitors are also to be tested. Figure 12 illustrates the fault-free signature when C_1 is set to 10pF and the signature obtained for a $+20\%$ deviation.

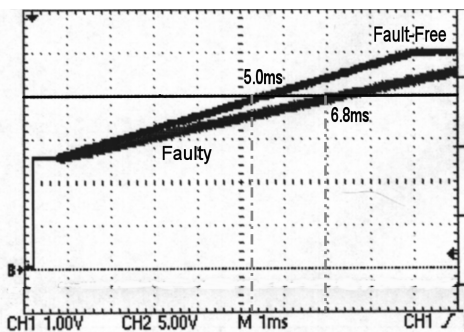


Figure 12: Time signatures for fault-free and $+20\%$ deviation

Figure 13 shows the results for the test of capacitances smaller than 30pF, plotted as a histogram. The signatures that deviate from the fault-free condition within a 0.3ms window are marked. This window represents the worst-case drifts for the fault-free time signatures observed in the performed experiments. These faults are considered undetected in this approach.

The same previously mentioned fault injection restrictions, which are due to the fixed values of programmable capacitors, hold here. Thus, excluding the 3 lowest capacitors into which faults cannot be injected, there are 59 capacitance values to be tested. As two of the four CABs may be tested concurrently (those that implement the oscillator), 118 TCs (2×59) are necessary to test this range.

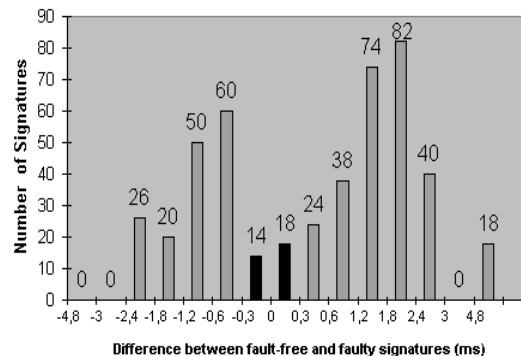


Figure 13: Histogram of the acquired time signatures for faults in capacitors smaller than 30pF

The best results of this test method are achieved for the third and last range of capacitances (31pF up to 62pF). For these values, the ORA is more sensitive to its own faults, allowing the four capacitors (two implementing the oscillator, and two implementing the ORA) to be tested concurrently. As there are 65 values in this last range, one needs to configure the device plus 65 times to fully test the capacitors of the device. For this range, results are plotted in the parametric curves shown in figures 14, 15, 16 and 17, respectively related to capacitors C_1 and C_2 of the oscillator, C_3 and C_4 of the ORA. Note that, for capacitors from 55pF up to 62pF (11 values), the faulty signature is not shown because it is not possible to inject $+20\%$ faults for this range. As it can be seen, deviations in the time signatures are large enough to be easily discriminated.

Table 2 summarizes the number of faults injected, covered or not, as well as the number of configurations needed in order to perform the proposed BIST strategy for each range of capacitances. It can be noticed that, for low values of capacitances, the architecture implemented for both the oscillator and the signature analyzer do not yield the best results. However, this difference may not occur with other devices, since the values of the programmable capacitors may be different, and lead to much better figures for the fault coverage.

3.2 Signature analysis for gain faults

For the gain testing using the ORA scheme, a different approach than the one presented in section 2.2 is adopted. Herein, we take advantage from the fact that the time signature given by the ORA structure is also dependent on the start-up time of oscillations. This parameter also shows good sensitivity to gain variations.

As for capacitor testing, care must be taken when setting the closed loop gain of the oscillator, since large values yield short start-up times, and the sensitivity of the time signature to the gain is decreased. On the other hand, low values yield large start-up times, which increase testing time significantly.

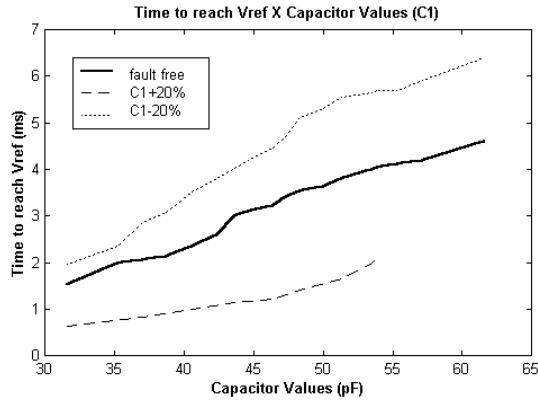


Figure 14: Fault-free and faulty signatures for capacitor C_1 in oscillator, for values larger than 30pF

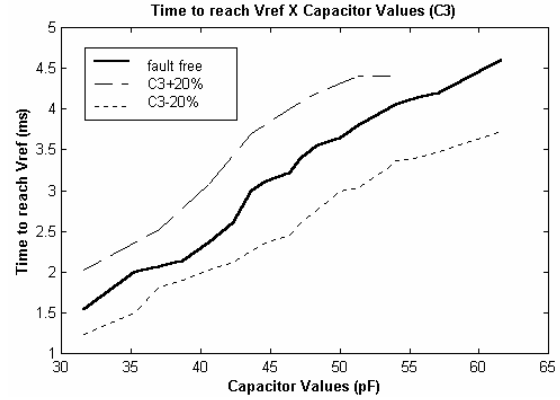


Figure 16: Fault-free and faulty signatures for capacitor C_3 in first ORA integrator, for values larger than 30pF

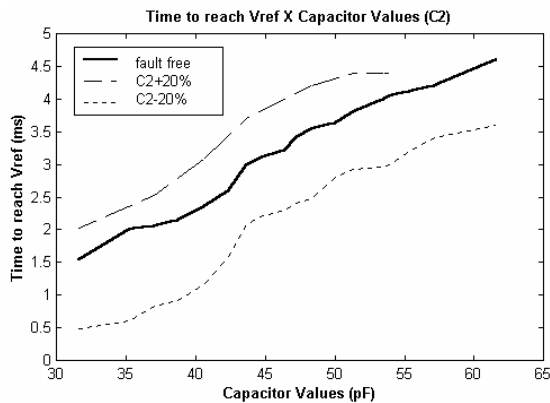


Figure 15: Fault-free and faulty signatures for capacitor C_2 in oscillator, for values larger than 30pF

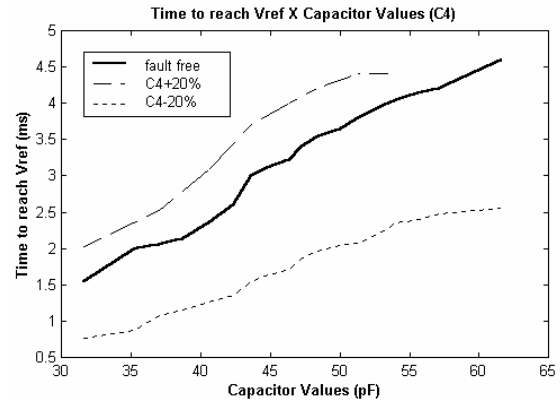


Figure 17: Fault-free and faulty signatures for capacitor C_4 in second ORA integrator, for values larger than 30pF

Table 2: Fault coverage evaluation and number of configurations to test capacitor using ORA approach

Capacitor Range	# Programm. Values	Injected Faults	Covered Faults	Not covered	Fault Coverage	Configurations
1pF to 30pF	62	464	432	32	93,1%	118
31pF to 62pF	65	476	476	0	100%	65
Total	127	940	908	32	96,6%	183

In order to test each IA of the oscillator, the programmable gain is varied all over the range, and the gain of the second IA in the loop is set in such a way that the modulus of the closed loop gain is larger than 7 and lower than 12. Note that one of the IAs shall have negative gain, otherwise oscillation will not be guaranteed. Also, the best trade-off between sensitivity and test time is found by setting the oscillator capacitors C_1 and C_2 to their maximum value. This way, without killing the oscillation, the signature sensitivity to the start-up time is ensured, and the testing time is still kept affordable. In the ORA, the gain of the IAs must be set to +1 and the capacitors of both integrators set to the same value of $C_3=C_4=61.59\text{pF}$.

If one uses the schematic of figure 10, all possible values of the IAs can be tested. Let the IA to be tested be IA1, and the second amplifier of the oscillator be IA3. Then,

considering the restriction on the overall closed loop gain, 10 configurations are necessary to test IA1, one TC for each possible positive value of gain. As this setup is also able to detect faults in the gain of IA3, there is no need to configure the device additional 10 times, but only 4, to cover the remaining gain values in IA3 that were not covered previously. Moreover, to test the negative values of gain, the same setup is used, but the signs are inverted. Hence, 28 TCs are necessary to test the pair of IAs in the oscillator loop. Since two of the IAs of the oscillator can be tested simultaneously with 28 TCs and there are four pairs of IAs in the studied device, it is necessary to configure the device 112 times for testing all the gains.

For the same fault model as in section 2.2 ($K_{\text{faulty}} = K \pm 1$), the obtained histogram for deviations in the gain value of all IAs is shown in figure 18. Differently from the OTS case, a 100% fault coverage is obtained here. The reason

is that ORA time signatures also take into account the oscillation start-up, while the OTS only measures signal frequencies and amplitudes.

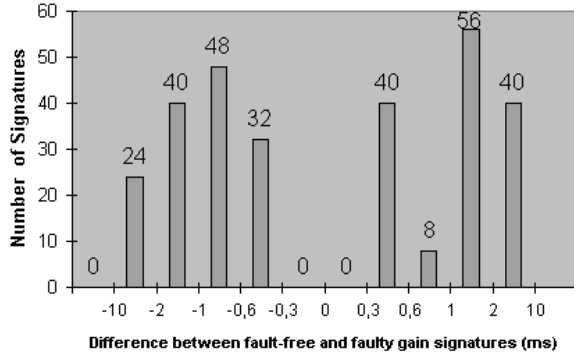


Figure 18: Fault-free and faulty signatures for IA gain values

3.3 Signature analysis for switch faults

According to table 1, there are two possibilities for the circuit to behave in case of faults in the switches: either it oscillates, or it does not. In order to discriminate whether the device is faulty or not, one must know how the switches are configured. For the test of S3, if it is stuck-on, then there is no oscillation and the ORA output is kept at the reference voltage of the FPAA device. In case of oscillation, a time signature is obtained. Thus, fault-free and faulty signatures are different and easily discriminated.

If non-oscillation is due to a S1 or S2 stuck-open, then the floating input of the oscillator CABs are susceptible to noise. The yielded time signature, in that case, does not present repeatability, given the random nature of noise. Thus, another setup is required.

Each CAB of the studied device is composed of two IAs, as it can be seen in figure 10. The implementation of the oscillator uses only one of these IAs. When testing the remaining switches, the one at the input of the IA that is not in the oscillator loop is the one under test. The gains of the first and second IAs in the loop are set to -10 and +1, respectively, so that from the output of the ORA a time signature is yielded. The IAs outside the loop are both set to +1. In case any of the input switches S1 or S2 outside the loop is faultily on, the overall gain of the oscillator is changed, and so does the time signature. Then, by evaluating these time signatures, it is possible to detect stuck-on faults at the switches. To test the switches that were in the oscillator loop in a previous TC, another configuration is performed, so that now they are outside the loop and thus can be tested.

To test stuck-open faults, a similar approach is adopted, in which one only needs to configure both switches S1 and S2 on. In case there is no stuck-open fault, there is a time signature which differs from that of a faulty situation, whatever the faulty switch is.

Finally, for the two oscillator CABs, 2 TCs are needed to test switch S3; to test the other switches, 1 TC is needed for stuck-open test and 2 TCs for the stuck-on test. To test the whole device (4 CABs), 10 TCs apply. All switch faults remain detected in the ORA approach.

4. Results Evaluation

Considering the results obtained for the ispPAC10 CAB testing, it is possible to estimate, for each method, the fault coverage for this device and the test application time, based on the number of required configurations and the FPAA programming time.

Table 3 shows the total number of reconfigurations for testing the different components of the CAB, as well as the fault coverage obtained in each situation. Comparing the results summarized in table 3, one can see that the fault coverage, in general, is better for the ORA method than it is for the OTS method, for the fault model considered. The latter has low fault coverage for testing the gain of the IAs, whereas the former presents some trouble in detecting faults in some ranges of capacitances. For the switches, both schemes can detect 100% of stuck-on and stuck-open faults.

Having the total number of configurations achieved in this test setup, knowing the time needed for the FPAA programming, and considering a minimum time to signal evaluation, it is possible to estimate the total testing time. Time for signal evaluation is that needed for the signal to settle and then be acquired and/or processed. For the OTS approach, this is equal to six full cycles of the output signal [10]. In average, a cycle of 40 μ s was obtained in our experiments with OTS and, in the ORA case, the average time signature found was about 8ms.

According to the manufacturer's manual [1], the reconfiguration time (programming and erasing) takes 200ms in the worst case. Thus, one can estimate the total test time as:

$$OTS : t_{test} = 172 \times [200ms + 6 \times 40\mu s] = 34.4s$$

$$ORA : t_{test} = 305 \times [200ms + 8ms] = 63.4s$$

Although the test time may seem high for both solutions, this is in part caused by the long time that the device needs to be programmed. In general, the testing time will depend on the complexity of the FPAA device and on its programming time. However, as one could intuitively conclude, the test time in the ORA approach is larger, as more configurations and a longer time are needed for signature evaluation. However, an external tester is still necessary in order to evaluate the test responses in the OTS method, and it must compute the frequency and amplitude of the two test outputs. For the ORA approach, the output evaluation hardware is very simple, consisting only of one comparator and one time counter. This way, the signature can be digitally encoded and compared with

the fault-free signature stored in a memory, for example, allowing the use of a very low-cost external tester. In addition, this is a great advantage if we consider a SoC environment, since resources already available in the system (such as a microcontroller) may implement such a simple tester and eliminate the need of an external equipment.

Another possibility is to merge both strategies into a single one, so that fault coverage, testing time and additional hardware requirements can be traded off in order to get a better performance with lower cost.

5. Conclusions

In this work, an important improvement to the FPAA test strategies presented in [4, 5] has been proposed, using the available resources of the device, not only to generate test stimuli, but also to analyze test responses. Herein, switch and gain faults were added to the fault model in [4, 5] that considered only capacitor deviations. Furthermore, the response analysis scheme in [5] was enhanced here for better measurement repeatability. Fault coverage, test time and external resource requirements were compared considering a pure Oscillation Test Strategy and a BIST approach based on programming an Output Response Analyzer into the FPAA.

A case study based on the FPAA ispPAC10 from the Lattice Semiconductor Corporation was shown and used to validate the test procedures proposed in the paper. Good fault coverage is obtained using both methods: the OTS covers faults on all capacitors, whereas the ORA presents better results for testing gains. Although the OTS approach is faster than the ORA, the latter does not require complex external hardware, allowing for a low cost test solution.

Nevertheless, the large number of TCs, thus long test time, is still an important issue. Test time can be decreased with new technologies that yield devices with faster programming phase, and also optimizing the proposed methodology so as to reduce the number of TCs. For example, test of switches and capacitors can be performed altogether, saving test application time.

Finally, more realistic fault models, as well as process variation issues, may be also included in future works. But, for that, further details on the physical implementation of the device under test must be made available by the manufacturers.

Although this study was directed to the Lattice FPAA family, the techniques applied here can be used to the test of other FPAA devices, since all of them have in their structure analog blocks that can be easily converted into oscillators and integrators.

Table 3: Fault coverage and number of TCs needed using OTS and ORA approaches

Component	Fault model	Injected Faults	Covered Faults		Not Covered		Fault Coverage		Configurations	
			OTS	ORA	OTS	ORA	OTS	ORA	OTS	ORA
Switches	Stuck_on/open	24	24	24	0	0	100%	100%	8	10
Gain	$K_{\text{faulty}}=K\pm 1$	288	144	288	144	0	50%	100%	40	112
Capacitors	$\pm 20\%$	940	940	908	0	32	100%	96.6%	124	183
Total		1252	1108	1220	144	32	88.5%	97.4%	172	305

6. References

- [1] *ispPAC Handbook*, Programmable Analog Circuits. Lattice Semiconductor Corporation, 2000.
- [2] B. Vinnakota (Ed.), *Analogue and Mixed Signal Test*. Prentice Hall, 1998.
- [3] M. Burns and G. W. Roberts, *An Introduction to Mixed-Signal IC Test and Measurement*. Oxford University Press, 2000.
- [4] T. Balen, M. Schreiber, F. Azaïs, M. Lubaszewski and M. Renovell, "OBIST Applied to FPAAs: A Case study", *4th IEEE Latin-American Test Workshop*, 2003, pp. 238-242.
- [5] T. Balen, A. Andrade Jr., F. Azaïs, M. Lubaszewski and M. Renovell, "An Approach to the Built-In Self-Test of Field Programmable Analog Arrays", in *Proc. 22nd IEEE VLSI Test Symposium*, 2004, pg. 383 – 388.
- [6] K. Arabi and B. Kaminska, "Testing Analog and Mixed-Signal Integrated Circuits Using Oscillation-Test Method". *IEEE Trans. CAD on Integrated CAS*, v.16, n. 7, July 1997.
- [7] G. Huertas, D. Vázquez, E. J. Peralias, A. Rueda and J. L. Huertas, "Testing Mixed-Signal Cores: A Practical Oscillation-Based Test in an Analog Macrocell". *IEEE D&T Computers*, v. 19, n. 6, pp. 64-72, 2002.
- [8] A. Andrade Jr., G. Vieira, T. Balen, M. Lubaszewski, F. Azaïs, M. Renovell, "Testing Global Interconnects of Field Programmable Analog Arrays", in *10th IEEE International Mixed Signals Testing Workshop*, 2004, pp. 231 – 236.
- [9] M. Lubaszewski, S. Mir and L. Pulz, "ABILBO: Analog Built-in Block Observer". In *IEEE/ACM International Conference on Computer-Aided Design*, 1996, pp. 600-603.
- [10] D. Vázquez, G. Leger, G. Huertas, A. Rueda and J. L. Huertas, "A Method for Parametric Extraction of Analog Sine-wave Signals for Mixed-signal Built-in Self-test Applications", *Design, Automation and Test in Europe Conference and Exhibition*, 2004, pp. 298-303.