Recycled FPGA Detection Using Exhaustive LUT Path Delay Characterization and Voltage Scaling

Md Mahbub Alam, Mark Tehranipoor, *Fellow, IEEE*, and Domenic Forte, *Senior Member, IEEE*

***Abstract*— Field-programmable gate arrays (FPGAs) have been extensively used because of their lower nonrecurring engineering and design costs, instant availability and reduced visibility of failure, high performance, and power benefits. Reports indicate that previously used or recycled FPGAs are infiltrating the electronics' supply chain and making the security and reliability of the critical systems and networks vulnerable. Current recycled integrated circuit (IC) detection procedures include parametric, functional, and burn-in tests that require golden or reference data. Besides, they are time consuming, require expensive equipment, and do not focus on FPGAs. In this article, we propose two recycled FPGA detection methods based on supervised and unsupervised machine learning algorithms. We develop a sophisticated ring oscillator (RO) design to exploit the degradation of lookup tables (LUTs) and use them in the proposed methods. In the supervised method, a one-class classifier is trained with RO frequencies, kurtosis, and skewness data obtained from unused FPGAs, which differentiates unused and aged FPGAs. The unsupervised method uses *k*-means clustering and Silhouette value analysis to detect suspect recycled components with very little (if any) golden information. In addition, we introduce a voltage scaling-assisted RO frequency measurement technique that improves the classification. The proposed methods are examined for Spartan-3A and Spartan-6 FPGAs, and the result shows that both methods are effective in detecting recycled FPGAs, which experience accelerated aging for at least 12 h equivalent to 70 days in real-time age.**

***Index Terms*— Counterfeiting, delay sensitivity, field-programmable gate array (FPGA) aging, lookup table (LUT) structure, ring oscillators (ROs) in FPGA.**

I. Introduction

SINCE their introduction, field-programmable gate arrays (FPGAs) have been widely used, and it is forecasted that the global FPGA market will reach approximately $10 billion by 2020 [10]. Reports from 2012 show that programmable logic is in the list of top five counterfeited electronic components with the percentage of 8.3% of reported counterfeit incidents [16]. With the increased volume of usage, FPGAs will likely become an even better target for counterfeiting, and thus, its reliability becomes a great concern to government and industry. In 2013, SMT Corporation estimated that recycled integrated circuits (ICs) comprised 80%–90% of all counterfeits in circulation worldwide [2]. The recycled electronic components are reclaimed or recovered from a system and then modified to be misrepresented as a new component of an original component manufacturer [3]. Since the recycled parts may have been exposed to harmful conditions, such as high temperature and high humidity, they are likely to have reliability issues. It should be pointed out that, given the vast volume of components recycled each year, it is even possible for the same component to be recycled multiple times. In today's complex electronic component supply chain, it is very challenging to prevent the infiltration of recycled FPGAs.

There have been a few works aimed at recycled IC detection by using electrical tests and on-chip sensors [6], [9], [11], [14]. Mainly, intrinsic delay [11] and path delay variations have been used [7], [21], [37] as sensors. All sensor-based-recycled IC detection methods incur hardware overhead, do not work for existing ICs in the market, and sometimes do not directly apply to FPGAs. Zhang *et al.* [6] proposed a path-delay-based method to detect recycled ICs that does not incur area overhead. Huang *et al.* [14] proposed statistical methods for detecting recycled ICs through the use of one-class classifiers and degradation curve sensitivity analysis. Bergman *et al.* [13] used power side channel to detect counterfeit components. These techniques rely on the drifts of the parametric profile of ICs and use authentic devices as reference points to detect recycled ICs. Most of the prior works on recycled IC detection neglected FPGAs and focused on ASICs.

In prior work, Dogan *et al.* [9] proposed a two-phase detection approach considering FPGA aging degradation that requires golden/reference data from known unused FPGAs. Besides, test FPGAs have to go through accelerated aging. We find that this method considered delay characteristics from a portion of the FPGA circuit, and thus, it is not effective to detect FPGAs with lesser prior usage. Moreover, the required golden/reference data may not be always available. For instance, it is hard to obtain data from known authentic and new FPGAs that are no longer being manufactured. In addition, unused FPGAs having accelerated aging suffer from early life degradation. Couch and Arkoian [8] used ring oscillator (RO) frequency of FPGAs and a machine learning-based classifier to identify the manufacturing lot by considering a lot-to-lot frequency variation. Here, test FPGAs were manufactured in comparatively old technology node

Manuscript received March 13, 2019; revised June 19, 2019; accepted July 15, 2019. Date of publication August 23, 2019; date of current version November 22, 2019. This work was supported in part by the National Science Foundation under Grant CCF-1423282 and in part by the National Institute of Standards and Technology under Grant 60NANB16D248. This work was presented in part at the IEEE International Test Conference (ITC), 2016. *(Corresponding author: Md Mahbub Alam.)*

The authors are with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32603 USA (e-mail: mahbub.alam@ufl.edu; tehranipoor@ece.ufl.edu; dforte@ece.ufl.edu).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVLSI.2019.2933278

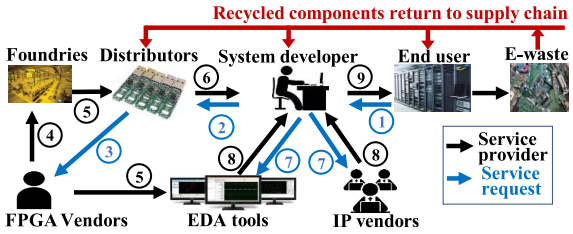

Fig. 1. Supply and demand market of FPGA with recycling threats. The numbers 1–9 show the order of services requested and provided by different entities.

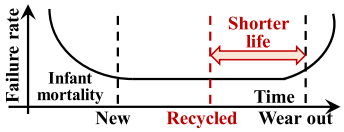

Fig. 2. Premature failure of recycled FPGAs.

(220 nm), where the impact of manufacturing variation is expected to be little.

In this article, we attempt to reduce the recycled FPGA detection failure rate while avoiding an accelerated aging phase entirely and propose effective detection techniques when there is little (if any) golden/reference data available. Our contributions are as follows.

1) We analyze FPGA aging to understand the nature of recycled components. A recycled FPGA is likely to have fully used, partially used, and unused lookup tables (LUTs) and each of them exhibits different aging behaviors in terms of path delay variation, which is useful for distinguishing new and used/recycled FPGAs.
2) We construct an exhaustive path delay characterization scheme by implementing a sophisticated RO using XNOR- and XOR-based mapping.
3) We develop the voltage scaling-assisted delay fingerprint that provides finer path delay difference between recycled and unused FPGAs.
4) We propose a supervised learning technique based on the support vector machine (SVM), which is trained on the path delay data features, i.e., mean, kurtosis, and skewness obtained from unused golden FPGAs.
5) We also propose an unsupervised machine learning method based on k-means clustering and silhouette value analysis, which requires a little reference data from golden FPGAs.
6) We provide silicon results of Spartan-6 and Spartan-3 FPGAs manufactured in different technology nodes to demonstrate the effectiveness of the proposed methods.

II. Problem Statement

A simplified model of supply and demand flow of FPGAs is shown in Fig. 1. End users except those who have their own hardware system development team give their system specification and design requirements to system developers. The system development process typically involves the acquisition of third-party intellectual property (IPs) and the use of licensed tools to generate the FPGA configuration bitstream file that defines the FPGA-based system based on users' requirements. Most systems are built on the existing FPGA chips in the market, which are obtained from distributors. Note that FPGA vendors might also act as distributors. When there is the need for specialized FPGA chip with dedicated IP components, customized FPGA chips can be fabricated normally through the FPGA vendors. The FPGA vendors include new IPs developed in-house or by third parties on their chips in order to better meet the requirements for emerging applications of end users. FPGA vendors provide EDA tools to system developer to support the design process. Also, FPGA vendors use third-party semiconductor foundries to fabricate the FPGA chips. Thus, new FPGAs enter into the market place, and end users or a system developer use them in their system.

Fig. 1 shows how the recycled FPGAs return to the supply chain. Counterfeiters reclaim FPGAs from e-waste or abandoned systems and sell them as new. These recycled components enter into the supply chain through unauthorized distributors and third-party system developers. As recycled components have prior usages, they face aging and impose severe reliability concerns. The bathtub is curve, as shown in Fig. 2, showing the early failures of recycled components because of prior usage. Here, recycled FPGA starts its lifetime (red dotted line) after a considerable amount of usage and has a shorter life span compared to the new component (black dotted line). If they are deployed in critical applications, such as aircraft engine control, nuclear reactor safety, and so on, catastrophic failure may occur. Early failures are also harmful to vendor reputation, e.g., Xilinx v. Flextronics [45].

In this article, we aim to quantify prior usages of FPGA and use this information to detect recycled FPGA components. For situations where delay information is available from reference/golden unused FPGAs, we propose a supervised method based on a one-class classifier as a decision function. The classifier provides a label for an FPGA purchased from a distributor as either used/aged or unused based on the delay information. For situations where there is a little to no information available from reference/golden unused FPGAs, we propose an unsupervised method that provides used/aged and unused labels. Note that the supervised method could be extended to multiclass classification in order to detect the approximate usages and portion of used LUTs across an FPGA. Both usage amount and localization of the used portion of FPGAs could be used to extend the lifetime of FPGA by reconfiguration, as shown in [46].

III. Background

A. Lookup Table Structure

LUTs act as configurable function generators and are considered as the basic building block of FPGA applications. Modern FPGAs allow modification to the mapped function of LUTs through reconfiguration. Thus, LUTs collectively implement billions of logic functions. It is thereby important to understand the behavior of LUTs to study the aging degradation for recycled FPGA detection. Generally, an n-input LUT contains 2^n number of SRAM cells to hold

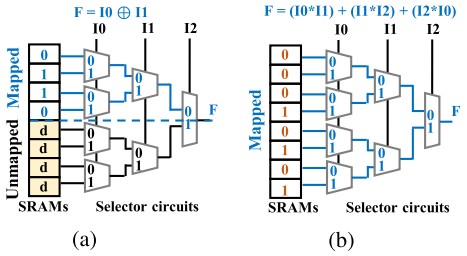

Fig. 3. LUT architecture and examples of logic mapping (blue color indicates used path). (a) Partially used LUT. (b) Fully used LUT.

the mapped values and a set of multiplexers to select the cells to drive out the SRAM cell value to the output. An example of an LUT architecture is shown in Fig. 3. In this example, a three-input LUT consists of eight SRAM cells and an 8:1 multiplexer. A tree of 2:1 multiplexers has been used to build the 8:1 multiplexer. Any three-input Boolean function can be realized by setting the truth table values in the SRAM cells, where the output is determined by the logic values of three-level hierarchical selectors ($I0$, $I1$, and $I2$).

B. Ring Oscillators' Configuration in FPGAs

ROs are implemented on FPGAs using LUTs, where they are mapped as an inverter. The propagation delay of such an inverter stage is the sum of the delay contributed by SRAM cells, selector transistors, and interconnect delay of the LUT.

C. Aging Mechanism

An operational FPGA slows down throughout its lifetime. The degradation mechanisms include bias temperature instability (BTI), hot carrier injection (HCI), time-dependent dielectric breakdown (TDDB), and electromigration [18]–[20], [22], [24], [33]. The BTI and HCI impose a significant impact on the switching speed of transistors. This impact is measurable and can be used for recycled FPGA detection.

1) Bias Temperature Instability: BTI is one of the major reliability concerns in MOS technology that affects the threshold voltage of transistors. Negative BTI (NBTI) and positive BTI (PBTI) increase the threshold voltage of pMOS and nMOS, respectively. pMOS transistors suffer from NBTI during prolonged times of gate-to-source negative bias stress. Prolonged stress creates interface traps at the interface of the gate oxide and channel that increase threshold voltage, which, in turn, decrease the switching speed. High temperature and voltages aggravate the impact of NBTI effect. The removal of gate-to-source stress allows the partial recovery of threshold voltage degradation. NBTI is dominant compared with PBTI beyond 65-nm technology nodes [33]. However, the introduction of high-k gate dielectrics and metal gate transistors elevates the effect of PBTI [19] that creates positive charge defects in nMOS transistors. Since modern FPGAs are scaling beyond 65 nm [4], they are likely to suffer from both PBTI and NBTI.

2) Hot Carrier Injection: Similar to BTI, the HCI effect leads to an increased threshold voltage and degrades the switching speed of the transistor. This phenomenon happens when electrons or holes in the substrate attain higher energies above the average due to a very high electric field in the drain region and get trapped in the gate oxide layer. Over time, charge defects build up an electric field within the dielectric layer, which leads to an increase in threshold voltage and a decrease in carrier mobility. It irreversibly slows down the switching activity of the transistor. In the lower technology node, the effective channel length gets smaller. As a result, degradation caused by HCI becomes worse.

D. Impact of Aging on FPGAs

BTI and HCI degrade the performance of FPGAs over time by affecting the threshold voltage of the circuits. The propagation delay of the BTI- and HCI-induced transistors increases. Hence, the selector circuit of LUT slows down with aging. The degradation of threshold voltage also alters the static noise margin of SRAM cells. Moreover, electromigration causes wire faults, and TDDB degrades the transistor performance. All degradation mechanisms are highly dependent on temperature. A significant amount of aging degradation has been reported in [30] and [42] at high temperature and voltages.

In this article, these aging phenomena are important for the following two reasons.

1) It motivates the need for the recycled FPGA detection method. The used FPGAs are slower than the new ones and could have unacceptably premature failures.
2) The fact that recycled FPGAs have measurable performance degradation implies that it is possible to detect them using electrical test.

IV. LUT Path Delay Analysis

A. LUT Types Based on Usage

1) Partially Used LUTs: The modern FPGA architecture provides four-, five-, or six-input LUTs. In most of the cases, all the LUT inputs are not needed to implement a logic function. Hence, many LUTs remain partially filled in a design. Karam *et al.* [17] mapped ten combinational logic benchmark circuits in six-input LUTs to analyze the amount of partially used LUTs in a design. The analysis showed that a significant amount of partially used LUTs exist across the diversity of benchmarks. Approximately 50% of the LUTs use four inputs or fewer and 82% of the LUTs use five inputs or fewer in combinational circuits containing less than 2000 LUTs. In large sequential benchmark circuits, we are supposed to find more partially used LUTs as reported in [17] showing 69% and 82% in the case of four inputs or fewer and five inputs or fewer, respectively. Although these numbers vary from application to application, a large amount of partially used LUTs are likely to be present in many applications.

We provide an example of adder implementation in Fig. 3(a) to show a partially used LUT. Here, a two-input "adder" is mapped in a three-input LUT using $I0$ and $I1$ inputs with the LUT contents of $(0, 1, 1, 0)$. The propagation paths of these

SRAM cells are depicted in blue color. This example uses four of the eight possible paths provided by three-input LUTs and leaves half of the portions unmapped (lower half of the LUTs) with do not care values. The amount of unmapped portions depends on the spare input. An LUT with one spare input leaves half of the LUT contents unused, and two spare inputs use a quarter of its resources. The paths in a mapped portion age differently due to switching activities of the input. The unmapped portions experience less aging [41] than mapped portions. This leads to an aging variation among the paths of partially filled LUTs.

2) Fully Used LUTs: All input pins are used in fully used LUT, as shown in Fig. 3(b). Here, all SRAM cells are mapped to implement a carry-out generator in a three-input LUT. The delay variations are likely to exist in fully used LUT since BTI- and HCI-induced degradation depends on the history of LUT configurations, input signal probabilities [19], and switching activities. LUTs with different configurations and different input signal probabilities have different aging-induced path delays.

3) Unused LUTs: FPGA capacity exceeds the average logic required for different applications due to the granularity of the available device sizes. A logic utilization of around 80% is typical in many applications. Thus, unused/spared resources are available in most of the cases. Trimberger [4] showed that FPGA resources exceeded the needs of low-end applications and moved on to high-end ones in a regular period. As FPGAs are more pronounced for low-end applications, unused resources are expected in many low-end designs. Aging degradation of these unused LUTs is less than that of used LUTs which go through different ac/dc stresses [41].

B. Necessity of Exhaustive Path Configurations

In [9], ROs are created by implementing inverter chains in LUTs. Since the inverter is a single input logic, it spreads over a single LUT path. The ROs implemented using one set of input pins do not cover all the paths of the LUTs. As described in Section IV-A, an application may comprise a different number of paths and their aging behavior differs. When a specific area of the FPGA is suffering from the aging, a method that only characterizes the delay of one path or some paths is not comprehensive enough to give a complete delay characterization. For instance, if paths contain spare LUTs or unused paths of partially filled LUTs, path delay will not represent the actual aging characterization. Therefore, a sophisticated test configuration is required to overcome this concern.

This article focuses on characterizing delay variations of all the possible paths regardless of unused, partially used, and fully used LUTs that help to better differentiate between unused and recycled FPGAs. In order to achieve this, we develop an RO that has the following properties.

1) All the MUX circuits of the LUT are used.
2) All the SRAM values are read out during test.

An RO with the above attributes provides all the possible paths for exhaustive delay characterization and requires to include all input LUT pins and SRAM cells in the design.

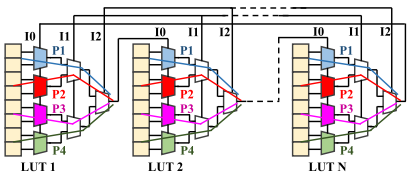

Fig. 4. RO in three-input LUTs with four possible paths (P1–P4) shown in different colors.

C. Implementation of Exhaustive Path Configurations

We implement ROs using an odd number of LUT stages and all the LUT input pins where each LUT stage acts as an inverter. Fig. 4 shows the formation of an RO with N odd number of stages using three-input LUTs. In this example, eight SRAM cells are located on the left-hand side in each LUT, and the output is selected by three-level hierarchical selectors (three-input MUX). The output of each LUT is connected to the input of the next LUT, and a closed loop is formed. Proper mapping of SRAM cell data and MUX selection bits is required to obtain the oscillation behavior. Since one of the inputs is required to form the chain, the rest of the input combinations can create different paths. For k-input LUTs, 2^{k-1} number of paths can be obtained [31].

For generalization, consider a k-input LUT with a total number of paths $P = 2^{k-1}$. We denote P paths as P_1, P_2, $P_3, \ldots, P_P$ and the frequency of each path as f_1, f_2, $f_3, \ldots, f_P$, respectively. Frequencies of a test RO configuration form the frequency array defined as

$$f_{\text{array}} = [f_1, f_2, f_3, \ldots, f_P]. \tag{1}$$

All the frequencies of f_{array} in a new FPGA will vary a little due to process variation [40]. The state-of-the-art design optimizes timing behavior of the LUTs, and thus, frequency variation is expected to be minimum in unused FPGAs. In used FPGAs, aging degradation will affect each path differently and create additional frequency variation.

Generally, designers do not have enough control over the synthesis, mapping, place, and routing tools that optimize Boolean functions for timing, routing, and area constraints. Thus, design tool causes issues in implementing a low-level design like RO that accesses all SRAM cells and selector paths. Hence, a Boolean function that implements inverter logic and selects all the SRAM cell entries of the LUT is needed. Standard logic XNOR and XOR act as an inverter when one of the inputs is considered as RO input and others are unchanged. The design tools cannot optimize these standard logics. Thus, XNOR- and XOR-based ROs satisfy the proposed RO requirement. Note that XOR tree-based test structures were used to detect faults and to perform built-in self-test in prior works [15], [35]. An example of XNOR- and XOR-based mapping for three-input LUT is shown in Table I. Here, four paths have been configured for inverter logic. A set of LUT input pins $\{I1, I2\}$ determine the paths P1–P4 while input pin $I0$ acts as input to inverter logic.

TABLE I

FORMATION OF PATHS USING XNOR LOGIC AND XOR LOGIC FUNCTION. HERE, IO AND F ARE THE INVERTER INPUT AND OUTPUT, RESPECTIVELY

LUT input			Inverter output (F)		Path
I2	I1	I0	XNOR	XOR	
0	0	0	1		P1
0	0	1	0		
0	1	0		1	P2
0	1	1		0	
1	0	0	1		P3
1	0	1	0		
1	1	0		1	P4
1	1	1		0	

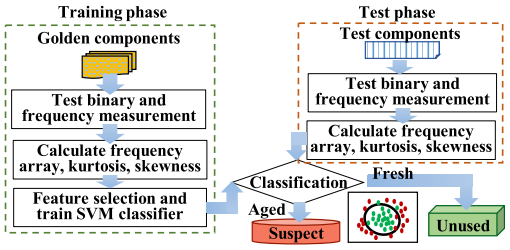

Fig. 5. Block diagram of the proposed supervised approach.

V. RECYCLED FPGA CLASSIFICATION

A. Classification With One-Class Support Vector Machine

We construct the frequency array of known golden/reference FPGAs and use these data to train a one-class SVM that creates a decision boundary. An FPGA under test goes through the same measurement process as the golden FPGAs. The trained one-class classifier examines test data using decision boundary and classifies test components as recycled or unused. The proposed approach is shown in Fig. 5.

An SVM is a distinctive classifying technique that has been used for data classification. This method separates the data into training and testing sets. The training set contains the class labels and several attributes (i.e., features). The features in our training set are the mean frequencies of LUT paths collected from golden (known unused) components, kurtosis, and skewness of the frequency distribution. Based on the training data set, the SVM creates a decision model that predicts the label of the test data given test data have the same attributes as the training data. Since we do not have the prior knowledge of the usage time and environmental condition of recycled components, we consider one-class SVM that only requires training data from a single class (i.e., the unused or golden FPGA frequency data). If test data are different, the decision model labels it as out-of-class (i.e., recycled in this article).

The one-class classification [39], [43] is an extension of the support vector methodology. This method can be imagined as a regular two-class SVM where the first class contains all the training data, and the origin is taken as the only member of the second class. Commonly used kernel functions include linear, polynomial, sigmoid, and radial basis function (RBF). In this article, RBF has been used as it offers fewer numerical difficulties [12]. Besides, it deals well with data that follow Gaussian distribution. Since the RO frequency data show a Gaussian distribution, the RBF kernel is the one we choose for our classification problem.

Before a classifier can assign labels to test components, the parameters of the classifier's model should be determined. To do so, we obtain data from M golden FPGAs. Each FPGA contains R number of ROs. Frequency vector of each sample is $f_{\text{array},1}, f_{\text{array},2}, f_{\text{array},3}, \ldots, f_{\text{array},R}$ for R number of ROs. Each element of the frequency array is constructed following the definition of the frequency array in (1). Later, the mean frequency of each LUT path is calculated from the frequency array and used as features to train the one-class classifier.

Location and variability of a data set play a significant role in the statistical analysis. In unused FPGAs, the frequency distribution is mostly affected by process variation. Thus, the shape of the frequency distribution is expected to be similar. On the contrary, in aged FPGAs, some paths age more than others, resulting in a change in shape and location of the frequency distribution. Kurtosis and skewness are the most commonly used metrics for location and shape variability of a data set. Kurtosis is a measure of whether the data are heavy-tailed or light-tailed relative to a normal distribution. In other words, kurtosis measures outliers present in the distribution. As the shape of the data changes with aging, the amount of outlier data will increase in the frequency distribution of a path. Thus, kurtosis will increase with aging. Skewness measures the lack of symmetry of the frequency distribution. Since LUT ages differently based on stress and signal probability, the frequency distribution of a path changes with aging. Skewness is used to take the symmetry variation into account.

Kurtosis for each LUT path is calculated as

$$k_p = \frac{\sum_{i=1}^{R}(f_{i,p} - \bar{f}_p)^4}{R \times \sigma^4} \tag{2}$$

where k_p is the kurtosis of the pth path of LUT, $\bar{f}_p$ is the mean frequency, σ is the standard deviation, and $f_{i,p}$ is the frequency of the ith RO of pth path.

Skewness for each LUT path is defined as

$$s_p = \frac{\sum_{i=1}^{R}(f_{j,p} - \bar{f}_p)^3}{R \times \sigma^3} \tag{3}$$

where s_p is the skewness of the pth path of LUT.

The training data set includes mean frequency, skewness, and kurtosis of each path and is defined as follows:

$$S_m = [\bar{f}_1, k_1, s_1, \bar{f}_2, k_2, s_2, \bar{f}_3, k_3, s_3, \ldots, \bar{f}_P, k_P, s_P] \tag{4}$$

where S_m denotes the feature vector of the mth golden sample. For M golden samples, the total training set F_{training} is

$$F_{\text{training}} = [S_1, S_2, S_3, \ldots, S_M]. \tag{5}$$

The training set creates a decision boundary using the one-class classifier and is used later for recycled FPGA detection.

For any test FPGA, S_t represents the test data for R number of ROs

$$S_t = [\bar{f}_1, k_1, s_1, \bar{f}_2, k_2, s_2, \bar{f}_3, k_3, s_3, \ldots, \bar{f}_P, k_P, s_P]. \tag{6}$$

The decision boundary labels test FPGA as unused or suspect recycled based on these test data. Test FPGA is considered

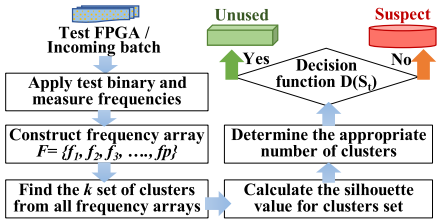

Fig. 6. Proposed approach using unsupervised machine learning.

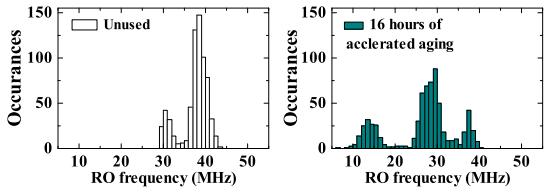

Fig. 7. Frequency distribution of unused and 16 h of aged paths in Spartan-3A FPGA.

as recycled if it is outside the decision boundary, otherwise unused.

B. Classification Using Unsupervised Method

The supervised detection method faces challenges when access to golden or reference FPGAs is limited. For example, legacy FPGAs are no longer being manufactured, but they are still in use in older systems. Thus, the necessity arises for a method that can detect recycled FPGAs with little or no prior knowledge. Here, we describe an unsupervised classification method for a single FPGA component and batch FPGAs, as shown in Fig. 6. Note that this method also utilizes frequency characteristics obtained from the XNOR- and XOR-based ROs. We perform clustering of the obtained frequency using the k-means algorithm. Then, for classification, we calculate the average silhouette value [36] of each cluster. The silhouette value for each observation point indicates the similarity of that point with other points in its own cluster compared with points in other clusters. We use this value to determine the appropriate number of clusters (ANC) for the FPGA under test. The number of clusters is primarily used as decision label (unused or recycled) for test FPGAs. Details of the detection process are described in the following.

As described earlier, for an unused FPGA, path characteristics are similar as only process variation impacts the delay. In used FPGAs, partially filled, completely filled, and spared LUTs age differently and increase the delay variation. Thus, the frequency distributions of aged and unused FPGAs are expected to be different. We present frequency distribution of unused and aged paths in Spartan-3A FPGA in Fig. 7. The paths are aged for 16 h using accelerated aging condition, as described in Section VII-A. The frequency distribution shows that frequencies are more clustered in unused FPGA while frequency distributions are scattered for aged paths. Thus, the frequency distribution of a used FPGA can be more multimodal since different paths of LUT aged differently, and thus, it can be divided into more clusters than unused FPGA. A similar phenomenon is also reported in [9].

We use the k-means method [23], [28] to partition frequencies into mutually exclusive clusters fixed *a priori* and to find the index of the cluster assigned to each frequency. Let us assume that test FPGA contains R number of ROs that provide R number of frequency array as defined in (1). Each RO contains P number of paths. Thus, the total number of frequencies used in this algorithm is the number of ROs (R) multiplied by the number of possible paths (P). Test data set is constructed as follows:

$$S_{\text{test}} = [f_{\text{array},1}, f_{\text{array},2}, f_{\text{array},3}, \ldots, f_{\text{array},R}]$$

where

$$f_{\text{array},i} = [f_1, f_2, f_3, \ldots, f_P], \quad i = 1, 2, 3, \ldots R. \tag{7}$$

The proposed method uses the k-means++ algorithm [26] for initializing the cluster centers, as it improves the quality of the final solution. This procedure incorporates batch and on-line algorithm [32] to solve the convergence problem by avoiding the global and local minima.

We calculate cluster indices of each frequency that indicates the cluster assignment of the corresponding frequency. Since our objective is to find the ANC for the given FPGA's data, a set of clusters $C = [2, 3, \ldots, K]$ is formed (i.e., two clusters from all data, and so on). The optimal choice of the number of clusters is determined by the silhouette value [36], which tells us how well the frequency data fit within its own cluster and differ from the neighboring clusters. The silhouette value SV_i for the ith frequency is expressed as

$$\text{SV}_i = \frac{OC_i - NC_i}{\max(OC_i, NC_i)}, \quad -1 < \text{SV}_i < 1 \tag{8}$$

where OC_i and NC_i are average Euclidian distance of the ith frequency point within its cluster and to neighboring cluster, respectively.

Next, we calculate the average silhouette values for each number of cluster set by finding the mean of the silhouette values of all the frequencies. Thus, for K cluster set, we have K number of average silhouette values. The ANC has a maximum average silhouette value. For suspect/recycled FPGA detection, we compare the ANC with a threshold number of clusters (TNC). The decision function can be expressed as

$$D(S_t) = \begin{cases} \text{suspect FPGA}, & \text{ANC of } S_t > \text{TNC} \\ \text{unused}, & \text{otherwise.} \end{cases} \tag{9}$$

The ANC is lower for unused FPGAs, as only process variation contributes to frequency clustering. On the contrary, used FPGAs experience variation due to aging and process variation and show a higher number in clusters. Thus, we can set a threshold value that is close to the expected number of clusters of unused FPGA. The threshold number can be determined from a few known unused FPGAs. The FPGAs from the same manufacturing batch/lot might be a good candidate for determining the threshold value of that batch/lot. Since FPGAs from the same manufacturing batch are not always available, we examine the impact of threshold variation on the false positive rate in Section VII and intend to find a threshold that minimizes false positives.

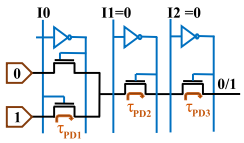

Fig. 8. LUT circuit portion in RO implementation.

VI. Voltage Scaling-Assisted FPGA Fingerprint

ROs exploit the intrinsic properties of the ICs. Generally, they generate noise and result in environmental variations to the surroundings. In addition, power supply variation plays a significant role in the propagation delay of FPGAs. Since this article captures the delay degradation due to aging, a high-precision measurement process is desired with minimum power supply and environmental noise. In this section, we search for an optimal measurement condition that minimizes the effect of noise and provides finer delay difference between unused and aged FPGAs.

The oscillation frequency of N inverter stages constructed in N LUTs can be expressed as

$$f = \frac{1}{2\sum_{i=1}^{N} \tau_{\mathrm{LUT},i}} \tag{10}$$

where $\tau_{\mathrm{LUT},i}$ is the propagation delay of the ith stage. The propagation delay of each LUT (τ_{LUT}) can be expressed as the sum of the delay of mux stages. Here, we consider that the delay of an SRAM cell is negligible. For the sake of simplicity, we consider a three-input LUT. Thus, the delay of an LUT stage is

$$\tau_{\mathrm{LUT},i} = \tau_{\mathrm{PD1}} + \tau_{\mathrm{PD2}} + \tau_{\mathrm{PD3}} \tag{11}$$

where $\tau_{\mathrm{PD1}}, \tau_{\mathrm{PD1}}$, and τ_{PD1} are propagation delay of mux stages 1–3, respectively. Fig. 8 shows active circuit in RO design using nMOS pass transistor-based LUT implementation and illustrates propagation delays τ_{PD}. For simplicity, we consider an nMOS-based structure to present the mathematical analysis. A similar analysis can also be deduced for transmission gate- and logic gate-based LUT [1] implementation.

The propagation delay of each mux stage (in general, termed $\tau_{\mathrm{PD},i}$) of an LUT has been calculated in Appendix I and is presented as

$$\tau_{\mathrm{PD},i} = \frac{C_L}{K}.\frac{V_{\mathrm{DD}}^2}{4}.\frac{\alpha+1}{(V_{\mathrm{DD}}-V_{\mathrm{th}})^{\alpha+1}} \tag{12}$$

where $K = \mu C_{ox}(W/L)$, V_{DD} is the supply voltage, V_{th} is the threshold voltage, and α is a constant that varies with technology node. The effect of supply voltage variation on this delay can be realized using sensitivity analysis. Let us consider the effect of delay variation ΔV_{DD} under nominal supply voltage V_{DD}. Since the state-of-the-art design optimizes the delay variation, $\Delta V_{\mathrm{DD}}/V_{\mathrm{DD}}$ is small and delay sensitivity ($S_{V_{\mathrm{DD}}}^{\tau_{\mathrm{PD}}}$) with respect to supply voltage can be defined as follows:

$$S_{V_{\mathrm{DD}}}^{\tau_{\mathrm{PD}}} = \lim_{\Delta V_{\mathrm{DD}}\to 0} \frac{\frac{\Delta\tau_{\mathrm{PD}}}{\tau_{\mathrm{PD}}}}{\frac{\Delta V_{\mathrm{DD}}}{V_{\mathrm{DD}}}} = \frac{V_{\mathrm{DD}}}{\tau_{\mathrm{PD}}}\frac{\mathrm{d}\tau_{\mathrm{PD}}}{\mathrm{d}V_{\mathrm{DD}}}. \tag{13}$$

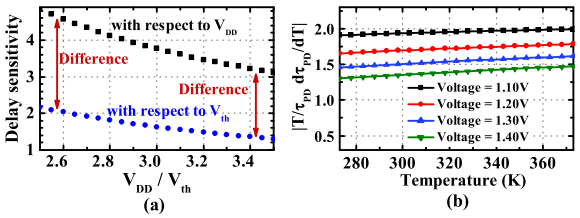

Fig. 9. Delay sensitivity with respect to (a) supply voltage and threshold voltage and (b) temperature.

The delay sensitivity of an LUT can be simplified as (details in Appendix II)

$$\begin{aligned} S_{V_{\mathrm{DD}}}^{\tau_{\mathrm{PD}}} &= \frac{\tau_{\mathrm{PD1}}}{\tau_{\mathrm{PD}}} S_{V_{\mathrm{DD}}}^{\tau_{\mathrm{PD1}}} + \frac{\tau_{\mathrm{PD2}}}{\tau_{\mathrm{PD}}} S_{V_{\mathrm{DD}}}^{\tau_{\mathrm{PD2}}} + \frac{\tau_{\mathrm{PD3}}}{\tau_{\mathrm{PD}}} S_{V_{\mathrm{DD}}}^{\tau_{\mathrm{PD3}}} \\ &\simeq -3\left[-2 + \frac{(\alpha+1).V_{\mathrm{DD}}}{V_{\mathrm{DD}}-V_{\mathrm{th}}}\right]. \end{aligned} \tag{14}$$

Recycled components are aged, and thus, their MOS transistors can be characterized by their degraded threshold voltages. This article exploits this degradation by measuring the delay of LUT stages. The delay sensitivity with respect to threshold voltage could, therefore, provide a good insight into the precision of RO measurement. We define sensitivity with respect to threshold voltage as

$$S_{V_{\mathrm{th}}}^{\tau_{\mathrm{PD}}} = \lim_{\Delta V_{\mathrm{th}}\to 0} \frac{\frac{\Delta\tau_{\mathrm{PD}}}{\tau_{\mathrm{PD}}}}{\frac{\Delta V_{\mathrm{th}}}{V_{\mathrm{th}}}} = \frac{V_{\mathrm{th}}}{\tau_{\mathrm{PD}}}\frac{\mathrm{d}\tau_{\mathrm{PD}}}{\mathrm{d}V_{\mathrm{th}}}. \tag{15}$$

Similar to (14), delay sensitivity with respect to threshold voltage is

$$\begin{aligned} S_{V_{\mathrm{th}}}^{\tau_{\mathrm{PD}}} &= \frac{\tau_{\mathrm{PD1}}}{\tau_{\mathrm{PD}}} S_{V_{\mathrm{th}}}^{\tau_{\mathrm{PD1}}} + \frac{\tau_{\mathrm{PD2}}}{\tau_{\mathrm{PD}}} S_{V_{\mathrm{th}}}^{\tau_{\mathrm{PD2}}} + \frac{\tau_{\mathrm{PD3}}}{\tau_{\mathrm{PD}}} S_{V_{\mathrm{th}}}^{\tau_{\mathrm{PD3}}} \\ &\simeq 3\frac{(\alpha+1)V_{\mathrm{th}}}{2(V_{\mathrm{DD}}-V_{\mathrm{th}})}. \end{aligned} \tag{16}$$

The delay sensitivity with respect to supply voltage and threshold voltage is shown in Fig. 9(a) by evaluating (14) and (16). We use following electrical parameters: technology node = 90 nm, threshold voltage $V_{\mathrm{th}} = 397$ mV, nominal supply voltage $V_{\mathrm{DD}} = 1.20$ V, technology parameter $\alpha = 1.17$, and $K = 243\ \mu\mathrm{AV}^{\alpha}$. Fig. 9(a) shows that the delay sensitivity due to threshold voltage variation is much lower than the sensitivity due to supply voltage variation. Thus, threshold voltage degradation in aged FPGAs might be suppressed by a large variation of the supply voltage. As a result, supply voltage variation makes the recycled FPGA detection process harder. We also see that the difference of delay sensitivity with respect to supply voltage and threshold voltage is higher at lower supply voltages and goes down with the increase of supply voltage. Thus, the delay becomes less sensitive to a threshold voltage and supply voltage variation. As a result, measuring delay at higher supply voltage might be beneficial for recycled FPGA detection.

Temperature variation is another critical parameter in delay measurement. Moreover, the on-chip temperature is expected to rise with the increment of the supply voltage. The delay sensitivity with respect to temperature can be used to find the effect of temperature variation during measurement. This

analysis can also be used to realize the effect of temperature variation on delay at different supply voltages.

The mobility and threshold voltage are temperature-dependent parameter in propagation delay equation. The mobility dependence on temperature can be approximated as [34]

$$\mu(T) = \mu_o \left[\frac{T}{T_o}\right]^{\alpha_u} \tag{17}$$

where T is the temperature, T_o is the nominal temperature, μ_o is the mobility at T_o, and α_u is an empirical parameter defined as the mobility temperature exponent (usually around 1.5). In a similar way, the threshold voltage dependence is given by

$$V_{\text{th}}(T) = V_{\text{tho}} + \alpha_{V_{\text{th}}}(T - T_o) \tag{18}$$

where V_{tho} is the threshold voltage at T_o, and $\alpha_{V_{\text{th}}}$ is the threshold voltage temperature coefficient with typical values around -2 mV/K [34]. The propagation delay as a function of temperature can be approximated as

$$\tau_{\text{PD},i}(T) = \frac{C_L T^{\alpha_u}}{\mu_o T_o^{\alpha_u}} . \frac{V_{\text{DD}}^2}{4} . \frac{\alpha + 1}{(V_{\text{DD}} - V_{\text{tho}} - \alpha_{V_{\text{th}}}(T - T_o))^{\alpha+1}} . \tag{19}$$

The delay sensitivity with respect to temperature is

$$S_T^{\tau_{\text{PD}}} = \lim_{\Delta T \to 0} \frac{\frac{\Delta \tau_{\text{PD}}}{\tau_{\text{PD}}}}{\frac{\Delta T}{T}} = \frac{T}{\tau_{\text{PD}}} \frac{\text{d}\tau_{\text{PD}}}{\text{d}T}. \tag{20}$$

A few steps of simple calculation lead to

$$S_T^{\tau_{\text{PD}}} \propto \frac{T \alpha_{V_{\text{th}}}(\alpha + 1)}{(V_{\text{DD}} - V_{\text{tho}} - \alpha_{V_{\text{th}}}(T - T_o))}. \tag{21}$$

Following (21), the delay sensitivity with respect to temperature variation is calculated at different temperatures and shown in Fig. 9(b). It shows that sensitivity increases slowly as temperature rises and drops significantly with the increase in supply voltage. Thus, it can be presumed that at any particular temperature (e.g., 300 K), supply voltage variation has more impact on sensitivity than temperature variation. Hence, minimizing the effect of supply voltage variation is more crucial for precise measurement, and this article introduces high-voltage measurement to obtain this goal. The above analysis is also applicable to other technology nodes.

VII. Experimental Results and Discussion

A. Experimental Setup

In this article, we implement XNOR- and XOR-based ROs on Spartan-3A (XC3S50A) and Spartan-6 (XC6SLX9) FPGAs appeared in Numato Lab FPGA development boards [44]. Spartan-3A (XC3S50A) and Spartan-6 (XC6SLX9) FPGAs are manufactured in 90- and 45-nm process technology node, respectively. FPGAs from two different technology nodes are investigated to examine the scalability of detection methods in terms of technology nodes. In Spartan-6 FPGAs, 350 ROs were created that cover 97% of the available LUTs. Each RO has 15 inverter stages (= 15 LUTs) with an additional enable logic (= 1 LUT). Spartan-6 has six-input LUTs that lead to 32 ($= 2^{6-1}$) possible paths for RO configurations. In addition, ROs are allowed to occupy some portion of the interconnect circuit that would add additional aging degradation. Hardmacros has been created for ROs to maintain the same internal routing and structure. Thus, we minimize the frequency variations induced by the routing differences.

TABLE II
Design of Experiment

Spartan-6		Spartan-3	
Benchmark	Occupied LUTs (%)	Benchmark	Occupied LUTs (%)
S9234	11%	S1423	27%
S13207	49%	S1488	33%
S35932	27%	S5378	42%
S38417	75%	S9234	84%
S38584	33%	S13207	65%

ROs are mapped to the FPGA in two measurement sessions in order to test all LUTs. This mapping is done by first placing hard macro ROs over a portion of the LUTs and measuring them. Next, we place ROs in remaining LUTs (not covered by the first set of measurements) and measure them. Measurements are taken at room temperature with the help of the onboard clock. Each frequency is measured ten times, and the average value is considered to mitigate the measurement noise. In Spartan-3A FPGAs, 112 XNOR- and XOR-based ROs were implemented, and each RO consists of seven inverter stages (=7 LUTs) and an enable logic (=1 LUT). Thus, 112 ROs cover 896 LUTs out of 1408. Other LUTs are used for measurement circuit, such as counter, data storage, transmission, and so on. The four-input LUTs in Spartan-3A FPGA resulted in eight ($=2^{4-1}$) paths. We measure RO frequencies at different supply voltages to find the best possible detection condition. A set of input supply voltages of [1.10 V, 1.15 V, 1.20 V, 1.25 V, 1.30 V, 1.40 V] are applied in this experiment, where the onboard nominal input voltage is 1.20 V for both FPGAs.

In this experiment, we intentionally age some FPGAs to create a recycled subset artificially. The aged FPGAs are subjected to accelerated aging (1.6 V and 125 °C) with ISCAS'89 benchmarks [27] for the various amounts of time, as shown in Table II. In Spartan-6 FPGA, benchmark circuits are repeated to obtain a specific amount of LUT utilization. This will help to understand the detection process at different amounts of utilization. In this article, 58 cases of Spartan-6 FPGAs are considered, where 18 of them were known unused and the other 40 were used for a different amount of time (4, 8, 12, and 16 h) using the benchmark circuits. In the case of Spartan-3A, the total number of FPGA was 32, where 12 of them were known unused and the rest were used for a different amount of time. We used a linear-feedback shift register (LFSR) to provide input signals to the benchmark circuits during accelerated aging. The FPGA development boards were bought as a batch. It can be assumed that their assembly process and electrical characteristics are similar.

ATS series thermostream has been used to generate high temperature. We use Keithley triple channel dc power supply to vary the internal supply voltage of FPGA. The estimated real-time aging of FPGAs is realized by thermal acceleration factor (taf) and voltage acceleration factor (vaf) described in

[29] and [42], which are generated by high temperature and supply voltage, respectively

$$\text{taf} = e^{\frac{E_a}{k}(1/T_{\text{op}} - 1/T_{\text{stress}})} \quad \text{and} \quad \text{vaf} = e^{\gamma\,(V_{\text{stress}} - V_{\text{op}})}.$$

The estimated acceleration factor is $145x$ (product of vaf $=$ 2.8x and taf $=$ 52x), where parameters are considered as activation energy $E_a = 0.5$ eV, Boltzmann's constant $k = 8.62 \times 10^{-5}$ eV/K, nominal operating temperature $T_{\text{op}} = 313$ K, stressed temperature $T_{\text{stress}} = 398$, voltage exponent factor $\gamma = 2.6$, nominal core voltage $V_{\text{op}} = 1.20$ V, and stressed core voltage $V_{\text{stress}} = 1.60$ V. Thus, accelerated aging time of $4, 8, 12$, and 16 h estimates 23, 46, 70, and 92 days of actual usage time, respectively.

We calculate the test time to provide a tentative test time required in the proposed methods. For the supervised method, developing a trained classifier requires measurement from unused FPGA. It is a one-time process and does not incur timing overhead during the test process. The time of test process includes programming test binary, measuring RO frequencies, and classification time. For an FPGA family, the programming time depends on programming circuitry such as USB-JTAG, SPI, and so on. We use USB-JTAG cable for programming, which requires 10 s. The measurement of 350 ROs requires 3.5 s with FPGA onboard clock of 100 MHz. A MATLAB software running in Intel Core i7-3.60-GHz processor has been used for prediction that takes less than 2 s to predict the status of an FPGA or to form the clustering. Overall, the time required to test an FPGA is less than 20 s. Thus, the proposed test process is fast. The hardware used in this process is memory storage, CPU, cable, and so on, which are inexpensive and easy to maintain.

B. Results for Supervised Machine Learning Algorithm

As described earlier, we measure RO frequencies of all Spartan-6 and Spartan-3A FPGAs. At first, we determined the features of the frequency distribution for the training and classification purposes. In both FPGAs, features from golden data are used to create a decision boundary for one-class classifier following the method described in Section V-A. Here, the detection performance of the classifier is illustrated in a receiver operating characteristic (ROC) curve. ROC graphs are commonly used in machine learning for visualizing, organizing, and selecting classifiers based on their performance [47]. It is a plot of the true positive rate against the false positive rate. One-class classifier yields a posterior probability or score for each FPGA that represents the degree to which an FPGA is a member of a class (unused or aged). The higher score indicates a higher probability of prediction accuracy. To obtain ROC curve for all test FPGAs, we sort the test instances by posterior probability/scores, from highest to lowest, move down the list, process one instance at a time, and update true positive and false positive rates as we go. An important property of the ROC curve is the area under the curve, which provides a probability of prediction accuracy. An area of 1 represents a perfect test; an area of 0.5 represents a worthless test. We use the area under this curve to compare the performances of classifiers at different supply voltages and age.

TABLE III

KURTOSIS AND SKEWNESS OF A SPARTAN-6 FPGA AT DIFFERENT AGING TIME AND VOLTAGES FOR S13207 BENCHMARK

Supply voltage	Kurtosis			Skewness		
	Time 0 hour	Time 12 hours	Time 16 hours	Time 0 hour	Time 12 hours	Time 16 hours
1.15	3.0351	3.1899	3.1295	0.3440	0.3485	0.3176
1.20	3.0267	3.0774	3.0659	0.2865	0.2279	0.2203
1.25	3.0594	3.2139	3.2761	0.2909	0.1067	0.0937
1.30	3.0570	3.5476	3.3201	0.2498	0.0173	0.0786
1.40	3.0210	4.9964	5.6634	0.2698	-0.4453	-0.5919

1) Classification With Kurtosis and Skewness Features: In this article, kurtosis and skewness are considered as features in contrast to [1]. In unused FPGAs, the frequency distribution is mostly affected by process variation. On the contrary, in aged FPGAs, some paths age more than others resulting in a change in shape and location of the frequency distribution. Table III shows the skewness and kurtosis values of Spartan-6 FPGAs aged with the s13207 benchmark. The skewness and kurtosis values were calculated using (2) and (3). It indicates that kurtosis and skewness value changes at different voltage levels and aging time with respect to unused condition. At unused condition, kurtosis and skewness are approximately 3.0 and 0.3, respectively. Kurtosis increases with aging time that means data get tailed heavily. These statistical behaviors are more vivid at higher supply voltages.

2) Comparison With Prior Work: Most prior works [38], [41] on FPGA aging do not focus on recycled FPGA detection. Thus, it is hard to measure the improvement achieved in this article. However, we compare our approach with prior work [1] where mean frequency of LUT paths was used as features. In this article, we include more statistical property of frequency data such as kurtosis and skewness in addition to the mean frequency of LUT paths. To show the improvement of classification, we compare the supervised classification of these two approaches. Classification using kurtosis and skewness features is shown in Fig. 10. We use a well-known performance comparison graph ROC to illustrate the performance of classifiers with and without kurtosis and skewness features. The area under the ROC curve presents the probability of successful detection. Fig. 10 shows that the successful detection rate increases significantly with the inclusion of kurtosis and skewness features. For instance, at 16-h aged devices, the probability of successful detection is approximately 91%, while it increases to almost 100% with kurtosis and skewness features. A similar improvement is also observed for 12- and 8-h aged devices. In the rest of the analysis, we use kurtosis, skewness, and mean frequency of each LUT paths.

3) Delay Sensitivity at Different Supply Voltages: We calculate the delay sensitivity to verify that voltage scaling helps in the detection process. From experimental results, we obtain the RO frequency and calculate the propagation delay using (10) for a Spartan-6 FPGA. Later, using these propagation delays, we calculate the delay sensitivity at different supply voltages with the help of (13). The delay sensitivities are 12.87, 8.11, 7.41, 5.86, and 5.58 for 1.15, 1.20, 1.25, 1.30, and

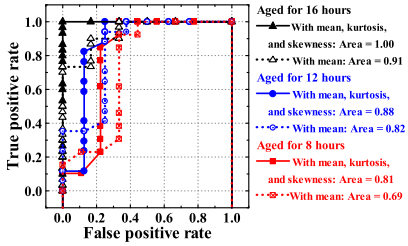

Fig. 10. ROC curve shows the detection rate at 1.30-V supply voltage.

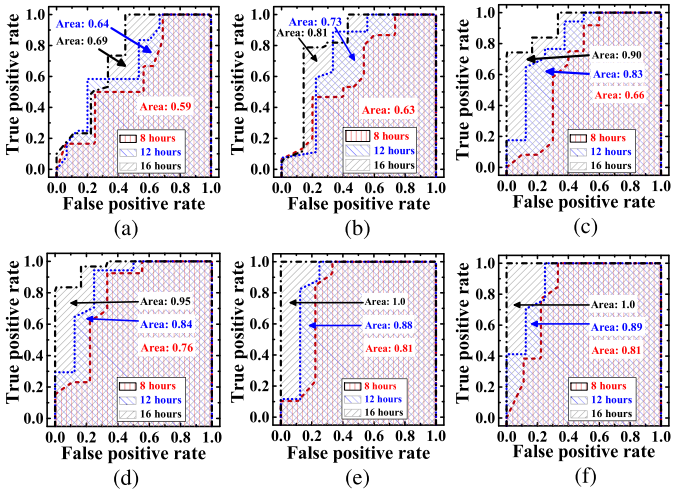

Fig. 11. ROC curves at different supply voltages from (a) to (f). (a) Voltage = 1.10 V. (b) Voltage = 1.15 V. (c) Voltage = 1.20 V. (d) Voltage = 1.25 V. (e) Voltage = 1.30 V. (f) Voltage = 1.40 V.

1.40 V, respectively. Thus, it can be concluded that ROs are less sensitive to environmental noise at higher supply voltages. We will also see that classifier performs well at supply voltages higher than nominal.

4) Classification at Different Supply Voltages: In the case of Spartan-6 FPGAs, we use 15 golden samples to train the classifier to obtain the decision boundary. Later, aged and unused devices undergo the detection process. The classifier detects all unused components. However, the classifier fails to detect FPGAs undergone through 4 h of accelerated aging.

Fig. 11 shows the classifier performance for 8, 12, and 16 h at different voltages. At each supply voltage, with the increase of aging, the detection rate increases. This scenario is expected as aging degradation increases with time. At nominal supply voltage of 1.2 V, the detection rate is 65% at 8 h, which increases to 90% for 16 h of aging. The classifier performance goes down at lower voltages (1.15 and 1.10 V). Even some 16-h accelerated aged FPGAs are detected as unused in the detection process and result in the lower detection rate of 69% and 81% at 1.10 and 1.15 V, respectively. This can be explained by considering the high-voltage sensitivity at lower supply voltages, i.e., the environmental noise has more impact on frequency measurement at low supply voltages.

It had been observed that supply voltages higher than nominal 1.20 V result in a better detection rate. With the increase of supply voltage, the classifier performance shows an increasing trend in aged FPGA detection. Classification at 1.30 and 1.40 V could detect most of the 16-h aged devices

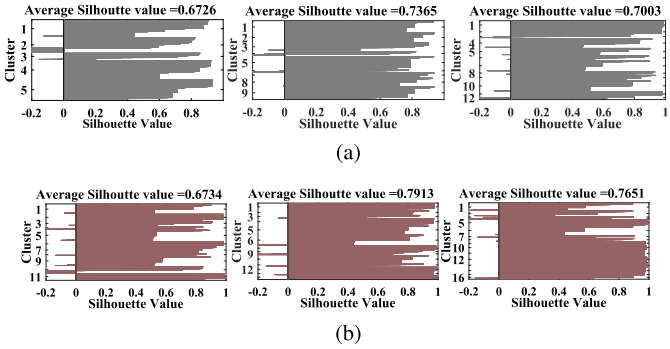

Fig. 12. Average Silhouette value at 1.4-V supply voltage. (a) Number of clusters: 6 (left), 10 (middle), and 12 (right) of unused FPGA. (b) Number of clusters: 11 (left), 14 (middle), and 16 (right) after 16 h of aging with S35932 benchmark.

while it could detect 95% instances at 1.25 V. In the case of 12 h of aging, the classifier could successfully identify aged or recycled devices in more than 80% cases. Thus, a lower delay sensitivity at high supply voltage enhances the recycled FPGA detection method. From all the ROC curves, it deems that 1.30 or 1.4 V might be good candidates for the optimum supply voltage.

We follow the same approach for Spartan-3A FPGAs and develop the one-class classifier by using ten FPGAs as golden/reference samples. After training, we feed the test FPGA data to the classifier. We consider 20 FPGA instances for 4, 8, 12, and 16 h of aging with each aging time contained five FPGAs. Like Spartan-6 FPGAs, the classifier could detect all unused FPGAs. It failed to identify all the 4-h aged FPGAs. The other test FPGAs were aged for 8, 12, and 16 h. The classifier could detect two of the FPGAs out of five after 8 h of aging at high voltages of 1.3 and 1.4 V. Since Spartan-3A FPGAs are fabricated in a lower technology node (90 nm) than Spartan-6, frequency degradation is lower. The performances of the classifier improve for 12- and 16-h aged devices. The classifier could detect four FPGAs as recycled after 12 and 16 h of aging. The FPGA with S1488 benchmark evaded the detection process. However, the size of s1423 is even smaller than s1488 and can be detected in the classifier. To find the reason, we focus on types of LUT and switching probability. We analyze the floor plan and find that s1423 uses more fully (four-input) used LUTs (195) than s1488 benchmark (171 LUTs). Aging degradation of these fully used LUTs is expected to be higher than unused and partially used LUTs. Besides, we use an LFSR to give random input to both benchmark circuits and thus calculate the switching probability of all nets of these benchmark circuits running under LFSR. We find that 60% of the nets of s1423 toggle less than 7%, while in case of s1488 benchmark, the number is less than 12% for the same percentage of nets. Thus, s1423 is expected to age more than s1488 in this work's implementation.

Based on the silicon results, it has been observed that the proposed method based on exhaustive path delay characterization detects recycled chips regardless of technology node with high probability, particularly for chips aged more than 12 h. This method uses golden data, and a higher number of golden

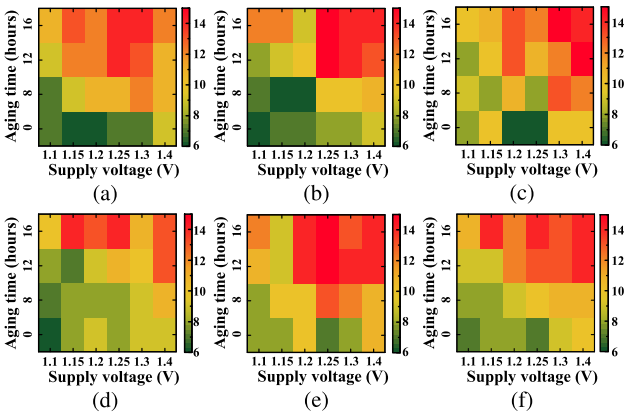

Fig. 13. Number of clusters of benchmark circuits at different voltages and aging time in Spartan-6 FPGAs. (a) s9234. (b) s13207. (c) s38417. (d) s38584. (e) s35932. (f) Average value.

chips will likely improve the performance of the classifier. In addition, the classifier needs to be upgraded for different lots/batches. Generating a universal classifier is not feasible, as fabrication processes drift from time to time.

C. Results for Unsupervised Machine Learning Algorithm

1) Calculation of the Number of Clusters: Fig. 12(a) and (b) shows an example of the average silhouette value calculation process for a Spartan-6 FPGA aged with S35932 benchmark at 1.40-V supply voltage. The silhouette value of each frequency is on the x-axis, and the number of clusters is shown on the y-axis. The maximum of the average silhouette value for the unused chip is 0.7365 in the case of ten clusters. Most of the frequencies in the ten-cluster scenario have a high silhouette value. Thus, they are well-matched within their clusters, and the ANC is ten. Similarly, we find 14 clusters for the aged chip with a maximum average silhouette value of 0.7913. The same calculation method was followed for the rest of the cases in Spartan-6 and also in Spartan-3A FPGAs.

2) Results of Spartan-6 FPGAs: In this analysis, the frequency data set for each FPGA is grouped into 2 to 16 number of clusters using the k-means clustering algorithm. Later, the ANC is selected following a maximum average silhouette value. Fig. 13 shows the ANC for benchmark circuits at different supply voltages and aging time. In this experiment, each benchmark circuit is implemented in five instances of FPGAs. We present the average value of clusters of each benchmark circuit in Fig. 13. The number of clusters for unused FPGAs is between 6 and 9 at supply voltages of 1.25, 1.20, 1.10, and 1.15 V. At higher supply voltages of 1.30 and 1.40 V, the number of clusters is in the range of 8–10. Since only process variation impacts the path delay, the number of clusters of unused FPGAs is similar.

With the aging time, the ANC increases for all the benchmark circuits. The aged devices have paths with different amounts of aging depending on the signal probabilities and the amount of occupied logic area. In this article, the number of cluster reaches to 14 at 16 h of aging time at high supply voltages for all the test benchmark circuits. For 12 and 8 h of aging time, the ANC follows a similar increasing progression. Although the increase in the number of clusters is small in the case of 1.10 and 1.15 V, it becomes greater at higher supply voltages of 1.25, 1.30, and 1.40 V. The average values of the number of clusters presented in Fig. 13(f) summarize the clustering trend with respect to aging time and voltages.

For recycled FPGA detection, a TNC is needed that provides maximum attainable distinction for any time of usage. The TNC is based on the delay property of FPGA. Thus, it is dependent on manufacturing technology, size, routing, and architecture of LUTs. Hence, it is tough to set a global or generic TNC for all FPGA families. Thus, the TNC needs to be set based on family, size, measurement, and so on. The TNC at each supply voltage can be selected based on the prior knowledge on reference unused FPGAs. In this regard, we varied the TNC and observed the true positive and false positive rates, as shown in Fig. 14. All 58 unused instances of Spartan-6 FPGAs and 40 aged FPGA cases are considered to calculate true positive and false positive rates. We consider two aging conditions (12 and 16 h) and supply voltages of 1.20 and 1.30 V to show the effect of the TNC on the false positive rate. In both the cases, the false positive rate increases with the increase of the TNC. Besides, the false positive rate decreases with the increase of aging time. At the nominal supply voltage of 1.20 V, a threshold of 11 clusters results in ∼100% true positive rate with the expense of 35% false positive rate for 16 h of aging. With the further increase of threshold number, the false positive rate increases. A threshold of ten clusters leads to 70% true positive rate and 25% false positive rate for 16 h of aged devices. The goal of this article is to identify recycled components. Thus, we want to set a threshold that maximizes the difference between the true positive rate and the false positive rate. For nominal supply voltages, 11 clusters seem to fulfill these criteria. However, the false positive rate is higher in the case of 12 h of aged devices. Thus, threshold eleven can be a proper choice for 16 or more hours of aged devices. A higher supply voltage provides an advantage in this regard [see Fig. 14(b)]. The false positive rate for 12-h aged devices reduces at the higher supply voltage of 1.30 V with 11 threshold clusters. In addition, it provides 100% true positive rate. The area under the curve is greater for 1.30 V, indicating better performance in classification. Note that recycled FPGAs are expected to age (a few months to years) way more than 16 h (92 days in real time) and we can expect that the FPGAs will have 11 or more clusters. Thus, false positives are expected to be lower than 20% considering 11 as the threshold. Similarly, the TNC can be selected for other supply voltages.

The number of reference unused FPGAs can be increased to add more confidence in setting a TNC. The result presented in this article shows that difference in the number of clusters is more explicit in 1.30 V than 1.20 V. Thus, higher voltages help in the detection process.

Since this method requires a little prior knowledge of unused FPGAs, it is more effective while testing lots/batches of FPGAs. In general, FPGA designers buy lots/batches of FPGAs for mass production. FPGAs within the same lot/batch are expected to show a similar number of clusters. Since

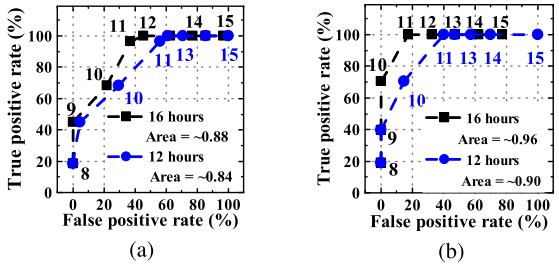

Fig. 14. ROC curves show the true positive and false positive rates at different TNCs for Spartan-6 FPGAs. Data label shows the TNC. (a) Supply voltage = 1.20 V. (b) Supply voltage = 1.30 V.

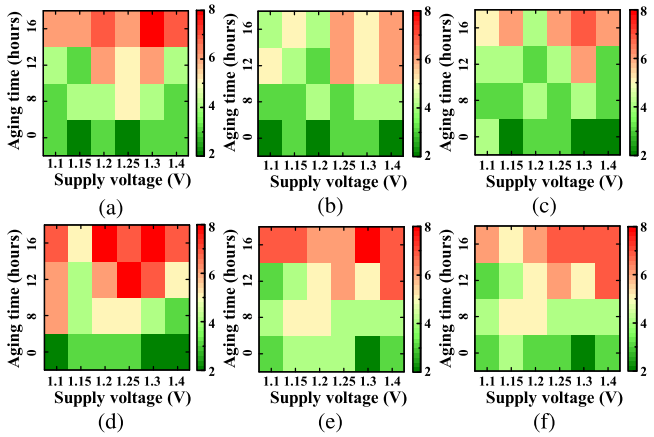

Fig. 15. Number of clusters of benchmark circuits at different voltages and aging time in Spartan-3A FPGAs. (a) s1423. (b) s1488. (c) s5378. (d) s9234. (e) s13207. (f) Average value.

recycled FPGAs are expected to age differently because of their usage variation, the number of clusters is expected to vary when recycled components are placed in a batch. Consider a lot/batch of FPGAs with unused and 70 days, 92 days, and more aged FPGAs. During the test process of this batch, the number of clusters will be in the range of 6–14. Thus, the whole batch can be considered recycled.

3) Results of Spartan-3 FPGAs: In Spartan-3A FPGAs, we consider frequencies from all eight possible paths to calculate the ANC. Similar to Spartan-6 FPGAs, at first, we form 2 to 8 number of clusters of the data using the *k*-means algorithm. Next, the ANC is calculated based on the maximum Silhouette value. Fig. 15 shows the number of clusters for different benchmark circuits and average values for all Spartan-3A FPGAs. Note that each benchmark circuit is implemented in multiple instances of FPGAs, and the average value of clusters is considered in this analysis. Like Spartan-6 FPGAs, at a nominal 1.20-V supply voltage and unused condition, FPGAs showed similar clustering nature, and here it is 3. We observe that the number of clusters in Spartan-3A FPGAs is comparatively smaller than Spartan-6 FPGAs. One probable explanation of this discrepancy is the fewer number of LUT paths and lower technology (90 nm) node of the Spartan-3A devices.

The ANC increases with the aging for the benchmark circuits specifically for larger benchmarks s9234 and s13207 that occupied 84% and 65% of the available LUTs, respectively.

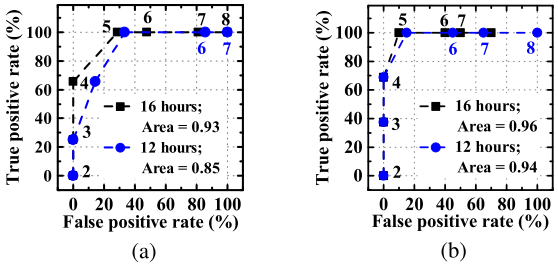

Fig. 16. ROC curves show the true positive and false positive rates at different TNCs for Spartan-3A FPGAs. Data label shows the TNC. (a) Supply voltage = 1.20 V. (b) Supply voltage = 1.30 V.

For most of the cases at 16 h, the ANC reaches to seven except s1488 benchmark where data are concentrated, and the number of clusters is low compared to others. Similar to Spartan-6 FPGAs, 16-h aged devices exhibit a higher increment in the ANC for most of the voltages. At higher supply voltages of 1.30 and 1.40 V, this difference is more noticeable as shown in Fig. 15(f).

In the case of 8 h, the number of cluster of aged chips is comparable to the unused chips. This indicates that the classifier based on unsupervised learning is less effective for 8-h aged devices. For 12 h of aging time, the ANC is higher than unused components, and it is more at higher voltages (1.30 and 1.40 V). The average value of the number of clusters plot [see Fig. 15(f)] indicates that the change in the ANC follows a similar trend of Spartan-6 FPGAs considering aging time and supply voltages.

Similar to Spartan-6 FPGAs, we vary the TNC to examine the false positive and true positive rates, as shown in Fig. 16. We consider 32 unused instances in this experiment. Since 20 samples were undergone aging, we consider their number of clusters while calculating the false positive rate. At 16 and 12 h of aging, the maximum difference between the true positive rate and the false positive rate is observed at threshold clusters of 5 for both supply voltages of 1.20 and 1.30 V. The false positive rate is 25% and 11% for 1.20 and 1.30 V, respectively, at 16-h aging. The area under the curve is greater for 1.30 V. Thus, it can be inferred that higher supply voltages improve the success rate of classification. Since recycled components are aged much more than 16 h (92 days in real time), the proposed unsupervised method is expected to identify recycled FPGAs with less false positives.

The unsupervised method used in this article performs better in case of Spartan-3A FPGAs than Spartan-6. This can be explained by considering the number of LUT paths. As the number of LUT paths is less in case of Spartan-3A (8 paths) than Spartan-6 (32 paths), frequency data are more clustered in Spartan-3A FPGAs.

VIII. Conclusion

In this article, we discussed a supervised method (with golden data) and an unsupervised method (no golden data required) for recycled FPGA detection. We categorized LUTs in partially used, fully used, and unused/spared LUTs based on the different path delay characteristics. Both methods char-

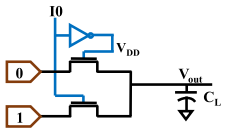

Fig. 17. Propagation of "0/1" through LUT in RO configuration.

acterize this path delay variation with the help of an advanced RO design that covers all possible LUT paths. We examined the proposed methods on 90- and 45-nm technology FPGAs. The experimental results show that the supervised method detects recycled FPGAs with the confidence of 90% after 12 h of accelerated aging time. In the unsupervised detection approach, the number of clusters shows an increasing trend with time. The performance of both methods improves at higher voltages than the nominal FPGA voltage. Both methods are fast and inexpensive since they do not require long assessment time or expensive hardware. Since it is based on machine learning and data labels, the supervised method is more appropriate when enough golden samples are available. The unsupervised method is better when the amount of golden samples is either very limited or nonexistent.

Appendix I
Propagation Delay of an RO Mapped in LUT

In this article, the inverter stage is implemented in an LUT. During oscillation, a portion of the LUT circuit structure is active as given in Fig. 8. A single mux stage with $I0$ input is shown in Fig. 17.

Here, C_L is the load capacitance that drives the rest of the LUT circuits. Generally, the delay of a transition can be evaluated by analyzing the charging/discharging of this load capacitance. For the sake of simplicity, the alpha-power law of current has been adopted [25].

Let us assume that V_{out} is initially charged to V_{DD}. To propagate through nMOS, a low-to-high transition at the gate terminal is applied. To find the high-to-low transition at load capacitance, we can express V_{out} with respect to transition time as

$$V_{\text{out}}(t) = V_{\text{DD}} - 1/C_L \int_0^t K(V_{\text{in}}(t) - V_{\text{th}})^{\alpha} dt \tag{22}$$

where K is a technology-dependent constant which is proportional to the transistor aspect ratio, V_{th} is the transistor threshold voltage, and α is a technology-dependent coefficient. Overall, the rise of the input $V_{\text{in}}(t)$ can be approximated by a saturated ramp with rise time T_r

$$V_{\text{in}}(t) = \begin{cases} V_{\text{DD}} t/T, & \text{when } t < T_r \\ V_{\text{DD}}, & \text{when } t > T_r. \end{cases} \tag{23}$$

After performing integration of (22) and considering the ramp behavior of input voltage $V_{\text{in}}(t)$, we get

$$V_{\text{out}}(t) = V_{\text{DD}} - \frac{K}{C_L}\frac{T_r}{V_{\text{DD}}}\frac{1}{(\alpha+1)}\left(V_{\text{DD}}\frac{t}{T_r} - V_{\text{th}}\right)^{\alpha+1}. \tag{24}$$

The propagation delay τ_{PD} is defined as the time difference between 50% input transition and 50% output level transition. Let us assume that input transition occurs at time $t = T_r/2$ and 50% output transition ($V_{\text{out}} = V_{\text{DD}}/2$) happens at time t_{out}. Thus, the propagation delay τ_{PD} is

$$\tau_{\text{PD}} = t_{\text{out}} - T_r/2. \tag{25}$$

From (24) and (25), propagation delay can be written as

$$\tau_{\text{PD}} = \frac{T_r}{V_{\text{DD}}}\left\{\left[\frac{(\alpha+1)C_L \cdot V_{\text{DD}}^2}{T_r 2K}\right]^{(1/\alpha+1)} - (V_{\text{DD}} - 2V_{\text{th}})\right\}. \tag{26}$$

Rise time T_r is unknown and Alioto and Palumbo [25] showed that it is twice of propagation delay in case of an RO. Thus, the above equation can be simplified to

$$\tau_{\text{PD},i} = \frac{C_L}{K} \cdot \frac{V_{\text{DD}}^2}{4} \cdot \frac{\alpha+1}{(V_{\text{DD}} - V_{\text{th}})^{\alpha+1}}. \tag{27}$$

Appendix II
Delay Sensitivity of an LUT

The LUT contains cascaded mux stages, as shown in Fig. 8. To find the delay sensitivity of an LUT, we need to take all the stages into account. Thus, the propagation delay of an n-input LUT is

$$\tau_{\text{LUT},i} = \tau_{\text{PD1}} + \tau_{\text{PD2}} + \cdots + \tau_{\text{PD}n}. \tag{28}$$

Following the definition of delay sensitivity with respect to supply voltage:

$$\begin{aligned} S_{V_{\text{DD}}}^{\tau_{\text{PD}}} &= \frac{V_{\text{DD}}}{\tau_{\text{PD}}}\frac{\mathrm{d}\tau_{\text{PD}}}{\mathrm{d}V_{\text{DD}}} \\ &= \frac{V_{\text{DD}}}{\tau_{\text{PD}}}\frac{\mathrm{d}(\tau_{\text{PD1}} + \tau_{\text{PD1}} + \cdots + \tau_{\text{PD}n})}{\mathrm{d}V_{\text{DD}}} \\ &= \frac{\tau_{\text{PD1}}}{\tau_{\text{PD}}}\frac{V_{\text{DD}}}{\tau_{\text{PD1}}}\frac{\mathrm{d}\tau_{\text{PD1}}}{\mathrm{d}V_{\text{DD}}} + \frac{\tau_{\text{PD2}}}{\tau_{\text{PD}}}\frac{V_{\text{DD}}}{\tau_{\text{PD2}}}\frac{\mathrm{d}\tau_{\text{PD2}}}{\mathrm{d}V_{\text{DD}}} \\ &\quad + \cdots + \frac{\tau_{\text{PD}m}}{\tau_{\text{PD}}}\frac{V_{\text{DD}}}{\tau_{\text{PD}m}}\frac{\mathrm{d}\tau_{\text{PD}n}}{\mathrm{d}V_{\text{DD}}} \\ &= \frac{\tau_{\text{PD1}}}{\tau_{\text{PD}}}S_{V_{\text{DD}}}^{\tau_{\text{PD1}}} + \frac{\tau_{\text{PD2}}}{\tau_{\text{PD}}}S_{V_{\text{DD}}}^{\tau_{\text{PD2}}} + \cdots + \frac{\tau_{\text{PD}m}}{\tau_{\text{PD}}}S_{V_{\text{DD}}}^{\tau_{\text{PD}n}}. \end{aligned}$$

The delay sensitivity for the power supply voltage is the sum contributions of each mux stage. Since these mux stages are close to each other, their transistor properties are expected to be very similar and, thus, result in similar propagation delay. Because of their locality, the corresponding gate delay sensitivity will be similar to all gates. As a result, the total sensitivity of a three-input LUT will be approximately three times of the sensitivity of a single mux stage.

References

[1] M. M. Alam, M. Tehranipoor, and D. Forte, "Recycled FPGA detection using exhaustive LUT path delay characterization," in *Proc. IEEE Int. Test Conf. (ITC)*, Nov. 2016, pp. 1–10.
[2] J. Villasenor and M. Tehranipoor, "Chop shop electronics," *IEEE Spectr.*, vol. 50, no. 10, pp. 41–45, Oct. 2013.
[3] M. Tehranipoor, U. Guin, and D. Forte, *Counterfeit Integrated Circuits: Detection and Avoidance*. New York, NY, USA: Springer, 2015.
[4] S. M. Trimberger, "Three ages of FPGAs: A retrospective on the first thirty years of FPGA technology," *Proc. IEEE*, vol. 103, no. 3, pp. 318–331, Mar. 2015.

[5] X. Zhang and M. Tehranipoor, "Design of on-chip lightweight sensors for effective detection of recycled ICs," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 22, no. 5, pp. 1016–1029, May 2014.

[6] X. Zhang, K. Xiao, and M. Tehranipoor, "Path-delay fingerprinting for identification of recovered ICs," in *Proc. IEEE Int. Symp. Defect Fault Tolerance VLSI Nanotechnol. Syst. (DFT)*, Oct. 2012, pp. 13–18.

[7] A. Amouri and M. Tahoori, "A low-cost sensor for aging and late transitions detection in modern FPGAs," in *Proc. 21st Int. Conf. Field Program. Logic Appl.*, Sep. 2011, pp. 329–335.

[8] J. Couch and J. Arkoian, "An investigation into a circuit based supply chain analyzer for FPGAs," in *Proc. 26th Int. Conf. Field Program. Logic Appl. (FPL)*, Lausanne, Switzerland, Aug./Sep. 2016, pp. 1–9.

[9] H. Dogan, D. Forte, and M. M. Tehranipoor, "Aging analysis for recycled FPGA detection," in *Proc. IEEE Int. Symp. Defect Fault Tolerance VLSI Nanotechnol. Syst. (DFT)*, Oct. 2014, pp. 171–176.

[10] *Consumer Electronics Drive FPGA Growth*, EPSNews, Mansfield, MA, USA, 2015.

[11] U. Guin, X. Zhang, D. Forte, and M. Tehranipoor, "Low-cost on-chip structures for combating die and IC recycling," in *Proc. 51st ACM/EDAC/IEEE Design Automat. Conf. (DAC)*, Jun. 2014, pp. 1–6.

[12] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A practical guide to support vector classification," 2003. [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf

[13] T. D. Bergman and K. T. Liszewski, "Battelle barricade: A nondestructive electronic component authentication and counterfeit detection technology," in *Proc. IEEE Symp. Technol. Homeland Secur. (HST)*, May 2016, pp. 1–6.

[14] K. Huang, Y. Liu, N. Korolija, J. M. Carulli, and Y. Makris, "Recycled IC detection based on statistical methods," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 34, no. 6, pp. 947–960, Jun. 2015.

[15] W. K. Huang, M. Y. Zhang, F. J. Meyer, and F. Lombardi, "A XOR-tree based technique for constant testability of configurable FPGAs," in *Proc. 6th Asian Test Symp. (ATS)*, Washington, DC, USA, Nov. 1997, pp. 248–253.

[16] *Press Room*, IHS, London, U.K., Apr. 2012.

[17] R. Karam, T. Hoque, S. Ray, M. Tehranipoor, and S. Bhunia, "Robust bitstream protection in FPGA-based systems through low-overhead obfuscation," in *Proc. Int. Conf. Reconfigurable Comput. FPGAs (ReConFig)*, Nov./Dec. 2016, pp. 1–8.

[18] S. Khan, S. Hamdioui, H. Kukner, P. Raghavan, and F. Catthoor, "Incorporating parameter variations in BTI impact on nano-scale logical gates analysis," in *Proc. IEEE Int. Symp. Defect Fault Tolerance VLSI Nanotechnol. Syst. (DFT)*, Oct. 2012, pp. 158–163.

[19] S. Kiamehr, A. Amouri, and M. B. Tahoori, "Investigation of NBTI and PBTI induced aging in different LUT implementations," in *Proc. Int. Conf. Field-Program. Technol.*, Dec. 2011, pp. 1–8.

[20] B. Khaleghi, B. Omidi, H. Amrouch, J. Henkel, and H. Asadi, "Estimating and mitigating aging effects in routing network of FPGAs," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 3, pp. 651–664, Mar. 2019.

[21] C. Leong, "Aging monitoring with local sensors in FPGA-based designs," in *Proc. 23rd Int. Conf. Field Program. Logic Appl.*, Sep. 2013, pp. 1–4.

[22] A. Lesea and A. Percey, "Negative-bias temperature instability (NBTI) effects in 90 nm PMOS," Xilinx Inc., San Jose, CA, USA, White Paper 224, 2005.

[23] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. IT-28, no. 2, pp. 129–137, Mar. 1982.

[24] D. Lorenz, G. Georgakos, and U. Schlichtmann, "Aging analysis of circuit timing considering NBTI and HCI," in *Proc. 15th IEEE Int. On-Line Test. Symp.*, Jun. 2009, pp. 3–8.

[25] M. Alioto and G. Palumbo, "Impact of supply voltage variations on full adder delay: Analysis and comparison," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 14, no. 12, pp. 1322–1335, Dec. 2006.

[26] D. Arthur and S. Vassilvitskii, "K-means++: The advantages of careful seeding," in *Proc. 18th Annu. ACM-SIAM Symp. Discrete Algorithms*, 2007, pp. 1027–1035.

[27] F. Brglez, D. Bryan, and K. Kozminski, "Combinational profiles of sequential benchmark circuits," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 1989, pp. 1929–1934.

[28] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statist. Probab.*, vol. 1, Oakland, CA, USA, 1967, pp. 281–297.

[29] R. Maes, V. Rozic, I. Verbauwhede, P. Koeberl, E. van der Sluis, and V. van der Leest, "Experimental evaluation of physically unclonable functions in 65 nm CMOS," in *Proc. ESSCIRC (ESSCIRC)*, Sep. 2012, pp. 486–489.

[30] A. Maiti, L. McDougall, and P. Schaumont, "The impact of aging on an FPGA-based physical unclonable function," in *Proc. 21st Int. Conf. Field Program. Logic Appl.*, Sep. 2011, pp. 151–156.

[31] M. Majzoobi and F. Koushanfar, "Time-bounded authentication of FPGAs," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 3, pp. 1123–1135, Sep. 2011.

[32] O. Mazhelis, "One-class classifiers: A review and analysis of suitability in the context of mobile-masquerader detection," *South African Comput. J.*, vol. 36, pp. 29–48, Jun. 2006.

[33] M. T. Rahman, F. Rahman, D. Forte, and M. Tehranipoor, "An aging-resistant RO-PUF for reliable key generation," *IEEE Trans. Emerg. Topics Comput.*, vol. 4, no. 3, pp. 335–348, Jul./Sep. 2016.

[34] S. Beer and R. Ginosar, "A model for supply voltage and temperature variation effects on synchronizer performance," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 11, pp. 2461–2472, Nov. 2015.

[35] M. Renovell, J. M. Portal, J. Figueras, and Y. Zorian, "RAM-based FPGA's: A test approach for the configurable logic," in *Proc. Conf. Design, Automat. Test Eur.*, 1998, pp. 82–88.

[36] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, pp. 53–65, Nov. 1987.

[37] M. Sadi, L. Winemberg, and M. Tehranipoor, "A robust digital sensor IP and sensor insertion flow for *in-situ* path timing slack monitoring in SoCs," in *Proc. IEEE 33rd VLSI Test Symp. (VTS)*, Apr. 2015, pp. 1–6.

[38] Y. Sato, M. Monden, Y. Miyake, and S. Kajihara, "Reduction of NBTI-induced degradation on ring oscillators in FPGA," in *Proc. IEEE 20th Pacific Rim Int. Symp. Dependable Comput.*, Nov. 2014, pp. 59–67.

[39] B. Schölkopf, R. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt, "Support vector method for novelty detection," in *Proc. NIPS*, 1999, vol. 12, pp. 582–588.

[40] P. Sedcole and P. Y. K. Cheung, "Within-die delay variability in 90 nm FPGAs and beyond," in *Proc. IEEE Int. Conf. Field Program. Technol.*, Dec. 2006, pp. 97–104.

[41] E. Stott, J. S. J. Wong, and P. Y. K. Cheung, "Degradation analysis and mitigation in FPGAs," in *Proc. Int. Conf. Field Program. Logic Appl.*, Aug./Sep. 2010, pp. 428–433.

[42] E. A. Stott, J. S. J. Wong, P. Sedcole, and P. Y. K. Cheung, "Degradation in FPGAs: Measurement and modelling," in *Proc. 18th Annu. ACM/SIGDA Int. Symp. Field Program. Gate Arrays*, 2010, pp. 229–238.

[43] D. M. J. Tax and R. P. W. Duin, "Support vector domain description," *Pattern Recognit. Lett.*, vol. 1999, pp. 1191–1199.

[44] (Jun. 2017). *Numato Lab*. [Online]. Available: https://numato.com/product-category/fpga-accelerated-computing

[45] EPSNews. (2014) *Another Wrinkle in Xilinx-Flextronics Suit*. [Online]. Available: https://epsnews.com/2014/01/28/another-wrinkle-xilinx-flextronics-suit/

[46] S. Srinivasan, P. Mangalagiri, Y. Xie, N. Viiaykrishnan, and K. Sarpatwari, "FLAW: FPGA lifetime awareness," in *Proc. 43rd ACM/IEEE Design Autom. Conf.*, Jul. 2006, pp. 630–635.

[47] T. Fawcett, "ROC graphs: Notes and practical considerations for researchers," *Mach. Learn.*, vol. 31, no. 1, pp. 1–38, 2004.