



**HAL**  
open science

## **Learning-Based Reliability Assessment Method for Detection of Permanent Faults in Clockless Circuits**

R. Aquino Guazzelli, Matheus Garay Trindade, Laurent Fesquet, Rodrigo Possamai Bastos

### **► To cite this version:**

R. Aquino Guazzelli, Matheus Garay Trindade, Laurent Fesquet, Rodrigo Possamai Bastos. Learning-Based Reliability Assessment Method for Detection of Permanent Faults in Clockless Circuits. *Microelectronics Reliability*, 2019, 100, <10.1016/j.microrel.2019.06.057>. <hal-03080316>

**HAL Id: hal-03080316**

**<https://hal.science/hal-03080316v1>**

Submitted on 21 Dec 2021

**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.



Distributed under a Creative Commons CC BY-NC 4.0 - Attribution - Non-commercial use - International License

# Learning-Based Reliability Assessment Method for Detection of Permanent Faults in Clockless Circuits

Ricardo A. Guazzelli, Matheus G. Trindade, Laurent Fesquet, Rodrigo P. Bastos

<sup>a</sup> Univ. Grenoble Alpes, CNRS, Grenoble INP\*, TIMA, 38000 Grenoble, France \* Institute of Engineering Univ. Grenoble Alpes

---

## Abstract

Permanent faults are a common issue of contemporary CMOS circuits during fabrication and its lifetime. In order to detect these faults, designers adopt logic and delay-oriented testing approaches to validate the circuit's structure. However, for clockless (i.e. asynchronous) circuits, applying well-known testing approaches is not trivial and bring undesirable overheads. To overcome that, the proposed method in this paper leverages the current signature properties of asynchronous circuits to detect permanent resistive-based faults. Through simulation experiments, we show the natural ability of asynchronous circuits in providing useful current signatures for identifying the presence of resistive-based faults. Moreover, the results demonstrate that our testing technique requires no extra circuitry or power ports to detect resistive-short faults  $\leq 3k \Omega$  and resistive-open faults  $\geq 70k \Omega$ .

---

## 1. Introduction

Advances in semiconductor manufacturing allowed the reduction of the minimum feature size of transistors and wires in the last decades. As a consequence, this has enabled a significant increase of the number of transistors in a single chip. However, as the transistor feature size reaches fundamental atomic limits, permanent faults that are trivial in previous generations become yield limiters for new technology nodes. Those permanent faults can occur not only during fabrication but also during the circuit's lifetime [1], e.g. aging and harmful radiation effects such as total ionizing dose (TID) [2]. If not properly detected, these faults pose a critical reliability risk to the circuit operation, implying that the circuit must be tested right after fabrication and periodically during its lifetime – especially if the circuit is intended to be used in a critical application.

Among the possible permanent faults, resistive-based faults are reported to be one of the most common faults in recent nanoscale technologies [3]. In order to test a Device Under Test (DUT) circuit and detect these types of faults, the literature presents approaches through delay and logic testing [4, 5]. Resistive-open faults are usually detected through delay testing [4], whereas resistive-short faults can be detected through both delay and logic testing [5]. Unfortunately, Process Variation (PV) in recent CMOS technologies reduces the effectiveness of delay testing for fault detection [4], compromising the reliability of the DUT.

Clockless circuits, also known as asynchronous circuits, have inherent reliable design features [6] thanks to the

use of delay insensitive encoding and local communication protocols instead of a global clock. Because of its reliable features, authors have also presented permanent fault diagnosis methods in asynchronous circuits [7, 8, 9, 10, 11, 12, 13, 14]. The literature presents adapted Design for Testability (DfT) techniques to detect permanent faults [7, 8, 9], permanent fault detection on system level (deadlock, handshake violation, out-of-sequence computation) [10, 11, 12, 13] and fault detection through quiescent current  $I_{ddq}$  testing [14].

The proposed diagnosis method in this paper deals with asynchronous circuits that intrinsically allow to individually analyze current signatures from different DUT stages, and thus better distinguishing fault-induced modifications on the current signature. We propose herein the use of transient current  $I_{ddt}$  (i.e. current signature) of asynchronous circuits to detect resistive-based faults, using a machine-learning algorithm called One-Class Support Vector Machine (OC-SVM). We show that it is only required measuring the global supply current  $I_{dd}$  from  $V_{dd}$  pin to obtain the individual current signatures of each DUT stage. The technique considers each point of individual current signatures, i.e. current points measured from different DUT stages, as an input feature to train OC-SVM and classify DUT samples. Thanks to the absence of a clock network and the local handshaking communication between the asynchronous DUT stages, a single input vector is not able to produce switching activity in idle stages. Hence each individual current signature carries only the switching activity of the active stage on that specific time frame, making the diagnosis of fault-induced modifications through the global current signature  $I_{ddt}$  easier. This diagnosis method can be adopted any time after fabrication,

---

\*Corresponding author: [ricardo.guazzelli@univ-grenoble-alpes.fr](mailto:ricardo.guazzelli@univ-grenoble-alpes.fr)

allowing to test the circuit right after fabrication and periodically through its field operation.

## 2. Resistive Faults in Integrated Circuits

This section presents a brief overview of the resistive faults covered in this work. It presents herein the adopted model of each fault and how its presence affects the current signature of a given circuit.

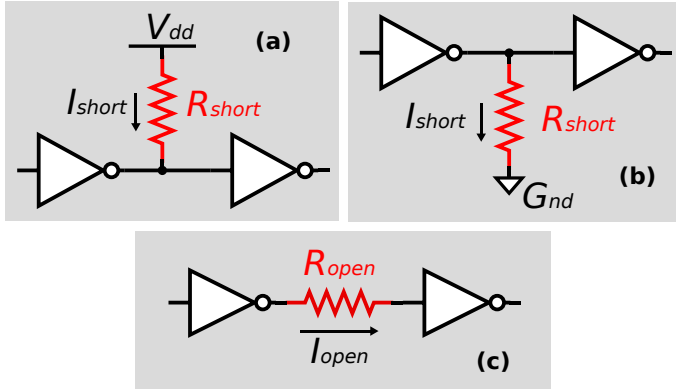


Figure 1: Circuit level models for resistive-short and open faults.

A resistive-short fault is a common fault in contemporary CMOS circuits [3]. This fault is usually caused by fabrication process, material, age or even packaging defects in the circuit. When present, a resistive-short connects circuit nodes that were not supposed to be connected, allowing the current flow an unintended path. Usually, these faults short transistor terminals themselves or supply lines ( $V_{dd}$ ,  $G_{nd}$ ). In circuit level, resistive-shorts are modeled as a parasitic resistor  $R_{short}$  connecting the unintended circuit nodes. Fig. 1 illustrates two examples of resistive-short faults connecting an internal logic node with a supply signal: (a) short to  $V_{dd}$  and (b) short to  $G_{nd}$ . In the former case, if  $R_{short}$  represents a significantly low resistance value, the left gate will not be able to drive its output to ‘0’ due to the current flow  $I_{short}$  from  $V_{dd}$ , generating a logic fault. For the latter, a logic fault can also appear with the presence of a low value of  $R_{short}$ , preventing the left gate to drive its output to ‘1’ as  $G_{nd}$  forces the node to ‘0’.

Open faults are defined as a defective connection between circuit nodes. Moreover, open faults can be classified as full opens and resistive opens [15]. Full opens are in the order of  $10\text{M}\ \Omega$  or higher resistances, while resistive opens have resistances lower than  $10\text{M}\ \Omega$ . Regarding testing, logic test usually targets full opens and delay test is more suitable to detect resistive opens. At a circuit level, resistive-opens can be modeled as a parasitic resistor between the circuit nodes that should be connected. As an example, Fig. 1 (c) illustrates a resistive-open fault  $R_{open}$  between two logic gates. In case the  $R_{open}$  is a full open fault, the left gate will not be able to drive properly its output, leaving the input of the following gate in

a floating state. If it is a resistive-open, the left gate is still able to drive the input of the following gate. However, the current limitation brought by the  $R_{open}$  affects the propagation delay of the left gate.

Due to the presence of these parasitic resistors, both resistive-short and open faults creates distortions in current signature during logic switching. If the fault distorts significantly the current signature, it is possible to distinguish the presence of a fault from process variations (PV).

## 3. Synchronous and Asynchronous Circuits

Synchronous circuit paradigm is predominant for digital design. Their fundamental assumption is that all components share a common notion of time guaranteed through a global clock network. Fig. 2 (a) illustrates a generic example of a linear 3-stage synchronous pipeline. A *clock* signal controls the memory elements  $R_0$ ,  $R_1$ , and  $R_2$  (typically flip-flops) and dictates when data propagates from one memory element to the other, through combinational logic blocks  $C_0$ ,  $C_1$ , and  $C_2$ . Differently from synchronous circuits, asynchronous circuits are implementable using one out of many different available design templates. Among them, authors cite that Quasi-Delay Insensitive (QDI) asynchronous circuits provide reliable and robust implementations against delay faults [16]. Fig. 2 (b) abstracts a linear 3-stage QDI asynchronous pipeline in which a local handshake scheme replaces the clock of synchronous circuits. Different from its synchronous counterpart, QDI asynchronous circuits combines memory elements with their respective combinational blocks – depicted as function blocks  $F_0$ ,  $F_1$  and  $F_2$ . QDI circuit data path *data* employs a multi-rail encoding, which allows the pipeline stages to detect the presence/absence of data, and acknowledgment signals *ack* indicate when a stage is available to receive data. Hence, QDI circuits are able to propagate and store data, independently of the delays of the function blocks.

### 3.1. Benefits of Asynchronous Current Signatures

In a synchronous circuit, the global *clock* signal normally controls several pipeline stages (e.g.  $S_0$ ,  $S_1$ , and  $S_2$  in Figure 2 (a)), switching all of them at every tick. If a single vector (herein token) stimulates the primary input  $data_0$  of the stage  $S_0$ , the global current signature  $I_{ddt}$  is influenced – during the first clock period – by the components  $I_{ddt0}$ ,  $I_{ddt1}$ , and  $I_{ddt2}$  (Figure 2 (c)), respectively from: the token activity in stage  $S_0$ ; and the switching activity of clock tree buffers and input circuitry of registers  $R_1$  and  $R_2$  in the idle stages  $S_1$  and  $S_2$ , which are both not computing any token. Clock-gating, power-gating, or even techniques for isolating the supply voltage of each stage with multiple supply pins are able to mitigate the current interference of components  $I_{ddt1}$  and  $I_{ddt2}$  at expense of additional hardware mechanisms. On the other hand, QDI circuits intrinsically cope with this issue by employing local handshake schemes.

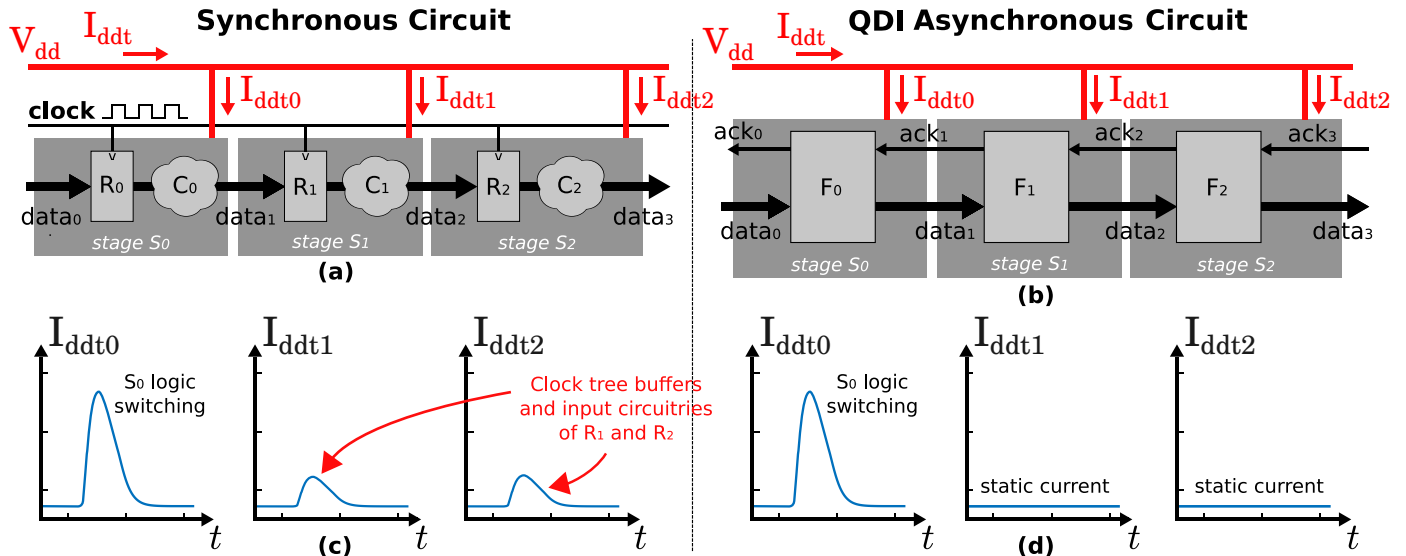


Figure 2: Example of a 3-stage linear pipeline: (a) synchronous and (b) QDI asynchronous circuits. The plots (c) and (d) represent the current signature of each pipeline stage in (a) and (b), respectively, during the propagation of a single input vector through the stage  $S_0$ . This example highlights the current influence that occurs in synchronous circuits. Even if only one stage is computing its inputs, the remaining stages still affects the total current signature.

Taking the same example as in (c), Figure 2 (d) illustrates the current signatures of each pipeline stage in (b). In this case, while  $S_0$  computes the input vector,  $S_1$  and  $S_2$  only contribute with static currents. The same applies if the token propagates further into the pipeline. When the token arrives at  $S_1$ , the previous stage  $S_0$  has already computed the token and remains in idle –  $S_2$  continues untouched. Next,  $S_2$  finally computes the token and  $S_0$  and  $S_1$  remain in idle, only contributing with static currents. As QDI circuits avoid the usage of a global clock, the influence from the clock tree is eliminated. Due to its local handshaking scheme, QDI circuits also avoid the influence of parallel switching activity of idle stages. These features provide a pipeline-level current signature isolation, making more significant any discrepancies caused by a permanent fault. However, in case the logic path employs fork structures, the propagated token generates parallel switching activity in all branches, implying the current signature comprises components from each branch.

## 4. Fault Diagnosis Method

The proposed method comprises seven main steps depicted in Fig. 3: (A) define stimulus procedure; (B) certify a set of fault-free samples; (C) extract fault-free current signatures; (D) OC-SVM training; (E) extract DUT current signatures and (G) identify faults on the DUT samples through OC-SVM classification.

### 4.1. Stimulus Procedure

The stimulus procedure is composed by a set of data tokens  $D$  that will be computed by the golden and DUT samples. Each data token  $d$  will be placed on the primary inputs at a time, i.e. a data token is only placed on the

primary inputs when previous one has already been computed and the logic is idle – there is no switching activity. Considering the aspects of the current signature on QDI circuits detailed in section 3.1, a data token only stimulates a single pipeline data path at a time, generating a current signature without the dynamic current components of other pipeline data paths. For a complete test, the stimulus procedure must contain a set of tokens that activate all the nodes in the circuit to test against any possible fault. Consequently, the testing time is a function of the delay to propagate a data token through a pipeline data path, multiplied by the number of tokens in  $D$ .

### 4.2. Golden Samples

The proposed diagnosis method requires a reliable reference to differentiate fault-free from faulty DUT samples. This reference comprises a set of fault-free ICs, i.e. golden samples. The current signatures of these golden samples are used to train a machine learning algorithm, further detailed in Subsection 4.4, that is capable of identifying distortions in the current signatures of DUT samples caused by faults. However, the minimum number of golden samples needed to train the SVM is highly dependent of the PV statistical distribution, as the algorithm needs to learn the characteristics of the PV.

### 4.3. Golden Current Signature Extraction

This step consists in reading the  $I_{ddt}$  of the golden samples, as indicated in Fig. 3, measuring the  $I_{ddt}$  current at fixed time steps since the insertion of a data token. According to subsection 4.1, by stimulating each golden sample with the data tokens  $d \in D$ ,  $|D|$  current signatures are extracted per sample. The current signatures are then grouped by the data token used to generate it, i.e. we

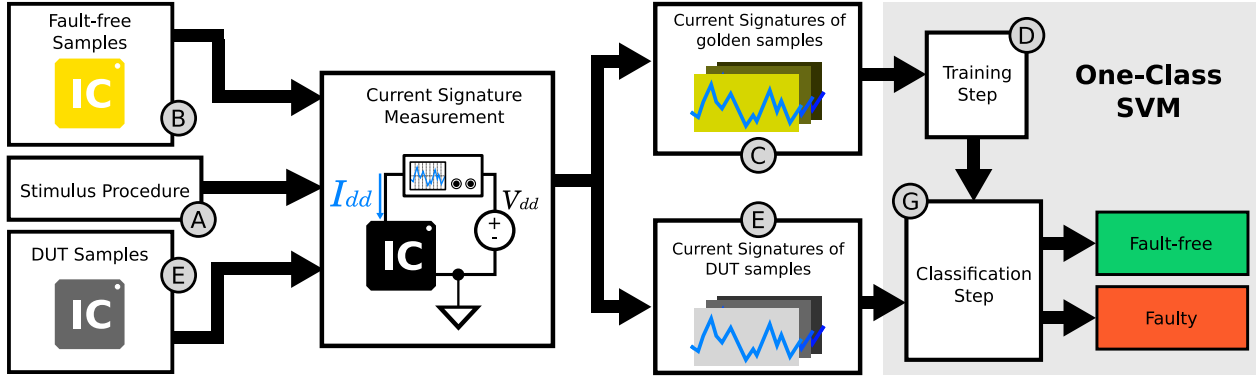


Figure 3: Proposed fault diagnosis flow highlighted in six main steps. Both fault-free and DUT current signatures are obtained with the same stimulus procedure and extraction method. While the OC-SVM training step considers fault-free samples to generate the classifiers, the OC-SVM classification step utilizes the DUT current signatures to classify them as fault-free or faulty.

generate the sets  $CS = \{CS_d \mid d \in D\}$ , where each  $CS_d$  is the set of current signatures of the golden samples when computing the data token  $d$ .

#### 4.4. OC-SVM Training

During the training phase, the OC-SVM algorithm learns the effects of PV on the current signature of golden samples, allowing the distinction between a fault-free sample from a faulty one. In our case, the algorithm only trains on the fault-free class, i.e. the available golden samples. The OC-SVM is an anomaly detection algorithm, as it trains only on one class and identifies discrepancies on new unseen data.

The diagnosis method considers each measurement of an extracted current signature as a training feature. For instance, if there are 50 golden samples and each current signature is composed by 30 current measurements, a OC-SVM training matrix will be sized 50x30, with the rows representing the golden samples and the columns representing the measurements, i.e. features. As presented in Subsection 4.3, the sets  $CS$  of current signatures are generated. The method then trains one classifier for each  $CS_d$ . For instance, if the stimulus procedure is composed of three data tokens, three OC-SVMs are trained, learning the idea of a fault-free current signature when a specific data token is propagated.

To find the necessary number of training samples for training the set of OC-SVM classifiers, the available golden samples are divided in two subsets: training and cross-validation subsets. As the name suggests, the training subset is reserved for training the machine learning algorithm, while the cross-validation subset is used to evaluate the accuracy of the classifier generated. The accuracy of the classifier for a given data token  $d$ ,  $Acc(d)$ , is defined in Equation 1, in which  $\#CorrClass\_cross\_val\_samples(d)$  is the number of correctly classified fault-free samples by the OC-SVM for token  $d$  and  $\#Cross\_val\_samples$  is the total number of fault-free samples in the cross-validation subset.

$$Acc(d) = \frac{\#CorrClass\_cross\_val\_samples(d)}{\#Cross\_val\_samples} \quad (1)$$

If the classification accuracy of at least one of the classifiers is not satisfactory, the number of fault-free samples in the training subset is increased. To have an idea of the accuracy independently of the fault-free samples used, i.e. independently of the training dataset, we estimate the mean of the classification accuracy over multiple trainings with different fault-free samples but maintaining a fixed size of the training subset.

#### 4.5. DUT Current Signature Extraction

This step takes the identical approach as section 4.3 and applies it to the DUT samples. The same stimulus procedure and current signature manipulation techniques are used to extract the data of the DUT samples. This will be used to evaluate the presence – or absence – of faults in the DUT samples. For each DUT, one current signature is extracted per data token  $d$ .

#### 4.6. OC-SVM Classification

Lastly, the classification step predicts the class of all DUT samples. The current signatures extracted from the DUT samples are evaluated in the set of OC-SVM classifiers generated in Subsection 4.4. Note that a current signature of a DUT sample for a data token  $d$  is only evaluated on the OC-SVM trained on  $CS_d$ . A DUT sample is said to be faulty if any of the classifiers indicate it as faulty. For example, if the stimulus procedure is composed of three data tokens, therefore each DUT sample has to be classified by three OC-SVM classifiers. If at least one of them classifies the DUT sample as faulty, it is considered overall as faulty.

## 5. Experimental Results

The experiments described in this section analyze the effectiveness of our technique for detection of resistive faults by inserting single faults inside an asynchronous study case.

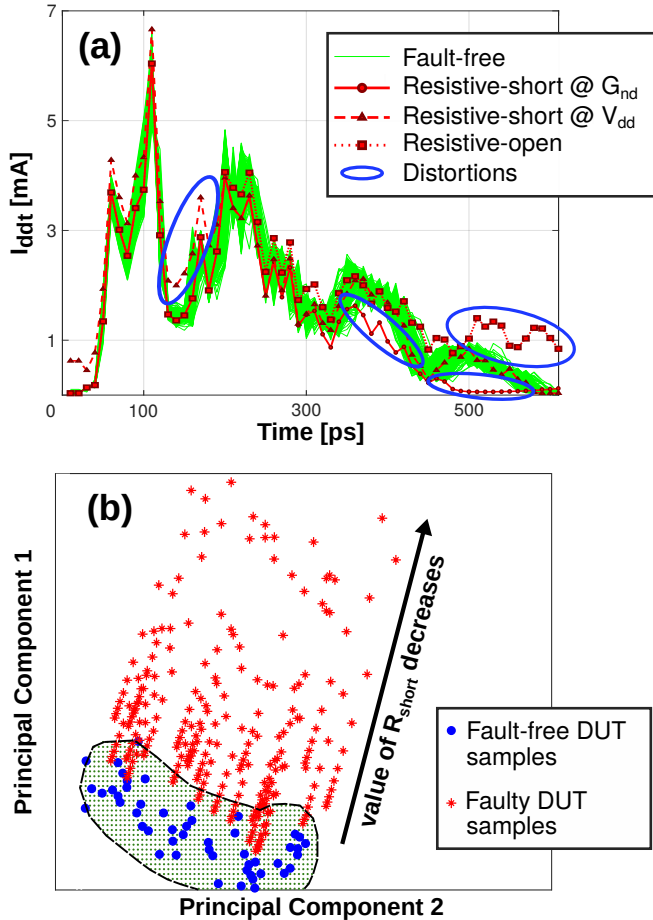


Figure 4: (a) Comparison of fault-free (green) and faulty (red) DUT samples. The ALU contains a single resistive-short fault in the third pipeline stage; (b) representation of an OC-SVM classifier of the ALU. Shaded area represents the generated classifier after Principal Component Analysis (PCA).

### 5.1. Experimental Setup

This work considers an 8-bit QDI asynchronous ALU unit [17] as study-case circuit. The ALU employs 13,974 transistors, three pipeline stages, dual-rail encoding and it has been designed in a FD-SOI 28-nm technology. The current signature data from fault-free samples have been represented by Monte Carlo (MC) simulations of all study-case circuits. Similarly to [18], we have considered three types of resistive faults: (1)  $R_{open}$ , (2)  $R_{short}$  to  $V_{dd}$  and (3)  $R_{short}$  to  $G_{nd}$ . These three faults have been scattered through the logic of the study-case circuit, in which each test case contains a single fault between the logic gate connections. The faulty test cases have been also simulated through MC simulations. Moreover, we simulated resistive faults varying from  $1k \Omega$  to  $100k \Omega$  at a temperature of  $25^\circ \text{C}$ . During simulation, a single data token has been propagated through the study-case circuit, stimulating the exact node where a fault had been inserted.

### 5.2. Analysis of Results

Fig. 4 (a) shows a comparison of extracted current signatures of the ALU during the propagation of a single data

token. At 0 ps, the data token is inserted in the primary inputs of the ALU, generating switching activity at the first pipeline stage. As the data token propagates through the ALU logic, the following pipeline stages are activated as well. When the data token is completely propagated, no switching activity is present and the current signature remains static. The light green curves represent the current signatures of fault-free samples and the three dark red curves show the current signature behavior of faulty DUT samples containing each one a different fault – one  $R_{short}$  at  $G_{nd}$ , one  $R_{short}$  at  $V_{dd}$  and one  $R_{open}$ . For this case, all faults are at the third stage of the ALU. Due to its location, the current signatures present distortions mainly between 400 and 600 ps, when the third stage of the ALU is switching its logic. The objective of the OC-SVM is to detect these distortions and flag the presence of a fault.

Next, Fig. 4 (b) shows an example of an OC-SVM classifier of the 8-bit ALU. Here, we show the perspective of the OC-SVM while classifying the current signatures. The shadowed area represents the classifier generated by the OC-SVM based on the data from the golden samples. The red asterisk points represent the data of DUT samples with  $R_{short}$  faults in the third stage of the ALU, while the solid blue points represent the data of fault-free DUT samples. Note that those fault-free DUT samples consist of a cross-validation subset of golden samples, meaning that this subset is different than the one used for the OC-SVM training. For the sake of illustration, the classifier dimensions have been reduced to two through Principal Component Analysis (PCA). The original data distribution has 61 dimensions and could not be displayed properly. PCA generates a reduced-dimension subspace to represent the data retaining the maximum possible precision represented on a targeted dimensionality – in this case, two dimensions. As the value of  $R_{short}$  decreases – the impact of the fault is higher – the distortion in the current signature is translated by a point further away from the classifier frontier.

For a better visualization of fault detection of the proposed technique, Fig. 5 (a) and (b) show the accuracy of our technique according to the values of  $R_{short}$  and  $R_{open}$ , respectively. The OC-SVM classifier is able to detect  $R_{short} \leq 3k \Omega$  and  $R_{short} \geq 70k \Omega$  with 100% accuracy. Regarding  $R_{short}$  faults, resistance values below  $5k \Omega$  compromise the capability of the logic gate to drive properly its output, generating a logic fault. On the other hand,  $R_{open} \geq 70k \Omega$  still allows the correct circuit operation at the cost of heavy delay overhead – QDI circuits can cope with this overhead. Considering our study case circuit, a faulty path with  $R_{open} = 70k \Omega$  presents a delay overhead of around 30% when compared to its fault-free counterpart.

Fig. 5 (c) presents the classification accuracy according to the number of device samples used for OC-SVM training. Similar to previous results, we consider cross-validation subsets to represent fault-free DUT samples. The training and classification steps were performed 100 times for each number of device samples for training. Con-

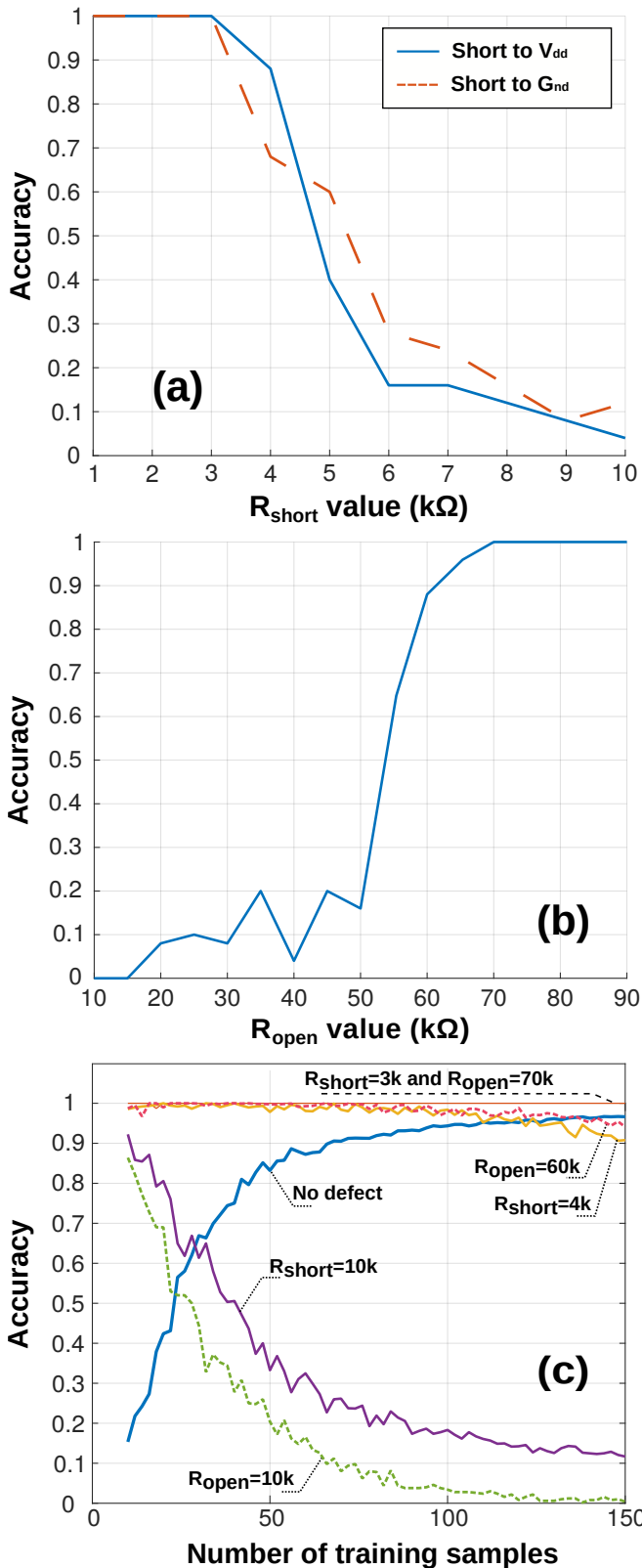


Figure 5: Fault detection accuracy of (a) resistive-short and (b) resistive-open faults according to the values of  $R_{short}$  and  $R_{open}$ , respectively; (c) Classification accuracy according to the number of device samples for OC-SVM training.

sequently, the accuracy results represent the mean values obtained from these 100 iterations. As expected, the higher the number of device samples for training, the higher is the capability of the OC-SVM to correctly classify fault-free DUT samples. This is highlighted by the blue line. In an under-fitting situation, the OC-SVM cannot distinguish current signature distortions caused by PV or a resistive fault, flagging most of the golden samples as faulty. Employing a higher number of device samples for training allows the OC-SVM to further learn how PV affects the current signature. With 100 and 150 device samples for training, the OC-SVM achieves accuracy of 95% and 97% while classifying fault-free DUT samples, respectively. Furthermore, by looking at the orange line at the top in Fig. 5 (c), we can see that the OC-SVM correctly classifies  $R_{short} = 3k \Omega$  and  $R_{open} = 70k \Omega$  no matter the number of device samples for training. The technique can achieve high fault-free detection without compromising the fault detection for  $R_{short} \leq 3k \Omega$  or  $R_{open} \geq 70k \Omega$ . On the other hand, the ability to cope with PV impacts significantly the detection of resistive-short faults with higher  $R_{short}$  and lower  $R_{open}$  values. Thus, we indicate cases in which the OC-SVM is not able to detect a fault: ( $R_{short} = \{4k, 10k\} \Omega$  and  $R_{open} = \{10k, 60k\} \Omega$ ). In these last cases, the distortion caused by the fault is still masked by PV, and the classification accuracy drops violently as the number of device samples for training increases. As the OC-SVM training employs higher number of device samples, the OC-SVM starts to learn the corners of the PV of the golden circuits, which may coincide with corners of PV faulty circuits, causing the accuracy to drop.

## 6. Conclusions and Future Works

This work proposes a non-intrusive diagnosis method for detection of permanent faults in QDI asynchronous circuits, revealing the advantages brought by their current signatures. By employing OC-SVM, our method is able to detect  $R_{short} \leq 3k \Omega$  and  $R_{open} \geq 70k \Omega$ . Our proposed method also provides high fault detection accuracy without requiring any extra circuitry into the DUT. As the method only requires the measurement of  $I_{dd}$ , the same testbench setup applied on regular post-silicon testing phase can be reused. Moreover, if  $I_{dd}$  monitoring is also available during the operation of the circuit, our method enables periodical field diagnosis.

As future works, we will explore other machine-learning algorithms, allowing the comparison between the fault detection accuracy of each algorithm. In addition, according to [18], setting different supply voltage and body bias levels in FD-SOI circuits allows better fault detection results, depending on the type of transistor employed. Hence, we expect that our method will provide more accurate results. Finally, the proposed method will undergo silicon validation using a fabricated test chip containing the study-case ALU used in this work.

## References

- [1] M. Abramovici, M. A. Breuer, A. D. Friedman, *Digital Systems Testing and Testable Design*, New York: IEEE, 1990.
- [2] E. G. Stassinopoulos, J. P. Raymond, The space radiation environment for electronics, *Proceedings of the IEEE* 76 (11) (1988) 1423–1442.
- [3] A. Karel, M. Comte, J. Galliere, F. Azais, M. Renovell, Influence of Body-Biasing, Supply Voltage, and Temperature on the Detection of Resistive Short Defects in FDSOI Technology, in: *IEEE Transactions on Nanotechnology*, Vol. 16, 2017, pp. 417–430.
- [4] S. Zhong, S. Khursheed, B. M. Al-Hashimi, Impact of PVT variation on delay test of resistive open and resistive bridge defects, in: *2013 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFTS)*, 2013, pp. 230–235.
- [5] M. L. Bushnell, V. D. Agrawal, *Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits*, Kluwer Academic Publishers, 2002.
- [6] Y. Monnet, M. Renaudin, R. Leveugle, Hardening techniques against transient faults for asynchronous circuits, in: *11th IEEE International On-Line Testing Symposium*, 2005, pp. 129–134.
- [7] C.-H. Cheng, J. C.-M. Li, An Asynchronous Design for Testability and Implementation in Thin-film Transistor Technology, in: *Journal of Electronic Testing*, Vol. 27, 2011, pp. 193–201.
- [8] K. Huang, T. Shen, C. Li, Test Methodology for Dual-Rail Asynchronous Circuits, in: *Design Automation Conference (DAC)*, 2017, pp. 1–6.
- [9] W. K. Al-Assadi, S. Kakarla, Design for Test of Asynchronous NULL Convention Logic (NCL) Circuits, in: *IEEE International Test Conference (ITC)*, 2008, pp. 1–9.
- [10] S. Peng, R. Manohar, Efficient failure detection in pipelined asynchronous circuits, in: *20th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'05)*, 2005, pp. 484–493.
- [11] C. LaFrieda, R. Manohar, Fault detection and isolation techniques for quasi delay-insensitive circuits, in: *International Conference on Dependable Systems and Networks*, 2004, 2004, pp. 41–50.
- [12] G. Zhang, W. Song, J. Garside, J. Navaridas, Z. Wang, Handling physical-layer deadlock caused by permanent faults in quasi-delay-insensitive networks-on-chip, *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 25 (11) (2017) 3152–3165.
- [13] N. Farazmand, M. B. Tahoori, Online detection of multiple faults in crossbar nano-architectures using dual rail implementations, in: *2009 IEEE/ACM International Symposium on Nanoscale Architectures*, 2009, pp. 79–82.
- [14] N. Nemati, P. Beckett, M. C. Reed, K. Fant, Clock-Less DFT-Less Test Strategy for Null Convention Logic, in: *IEEE Transactions on Emerging Topics in Computing*, Vol. 6, 2018, pp. 460–473.
- [15] R. R. Montanes, J. P. de Gyvez, P. Volf, Resistance characterization for weak open defects, Vol. 19, 2002, pp. 18–26.
- [16] A. Martin, M. Nystrom, Asynchronous techniques for system-on-chip design, *Proceedings of the IEEE* 94 (6) (2006) 1089–1120. doi:10.1109/JPROC.2006.875789.
- [17] L. A. Guimaraes, T. F. de Paiva Leite, R. P. Bastos, L. Fesquet, Non-intrusive Testing Technique for Detection of Trojans in Asynchronous Circuits, in: *Design, Automation Test in Europe Conference (DATE)*, 2018, pp. 1516–1519.
- [18] A. Karel, F. Azais, M. Comte, J. Galliere, M. Renovell, K. Singh, Comprehensive Study for Detection of Weak Resistive Open and Short Defects in FDSOI Technology, in: *2017 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, 2017, pp. 320–325.