Journal of Electronic Testing (2023) 39:27–40
https://doi.org/10.1007/s10836-023-06044-z

Identifying Resistive Open Defects in Embedded Cells under Variations

Zahra Paria Najafi-Haghi[1] · **Hans-Joachim Wunderlich**[1]

Received: 8 August 2022 / Accepted: 18 January 2023 / Published online: 9 February 2023

Abstract

Small Delay Faults (SDFs) due to weak defects and marginalities have to be distinguished from extra delays due to process variations, since they may form a reliability threat even if the resulting timing is within the specification. In this paper, it is shown that these faults can still be identified, even if the corresponding defect cell is deeply embedded into a combinational circuit and its observability is restricted. The results of a few delay tests at different voltages and frequencies serve as the input to machine learning procedures which can classify a circuit as marginal due to defects or just slow due to variations. Several machine learning techniques are investigated and compared with respect to accuracy, precision, and recall for different circuit sizes and defect scales. The classification strategies are powerful enough to sort out defective devices without a major impact on yield.

Keywords Resistive opens · Variations · Reliability · Machine learning

1 Introduction

Resistive opens and resistive bridges result often in SDFs [1, 2, 3] which are hard to detect during production testing. Since they may evolve rather early in the circuits lifetime and turn into catastrophic faults, they have to be covered during the test of high-quality systems [4, 5, 6]. It is well known that testing at varying voltages and especially low voltage testing increase the fault coverage significantly [7, 8, 9, 10], and modern systems with Adaptive Voltage Frequency Scaling (AVFS) have all the means to support this test strategy [11]. However, technology scaling comes with the additional difficulty that circuits are subject to process variations, which similarly affect the timing as SDFs do. In both cases, the circuit behavior may be slowed down but stays still within the specification. Yet, a slow circuit due to variations may be safe whereas a slow circuit due to resistive weak defects may form a reliability threat. Distinguishing cells slow due to variations from defective cells has been the subject of ongoing research [10, 12–14] and [15]. A severe limitation of the previous works comes with the fact, that the defective cells are subject to variations too. In some cases, a defective fast cell may still be faster than a defect-free one.

Process-induced variability due to imperfect fabrication process is unavoidable and has an impact on transistor attributes such as length, width, and oxide thickness. Means such as AVFS are exploited to countermeasure the effects of process-induced variability, and the emerging FinFET technology reduces its impact to some degree, but cannot suppress it entirely.

The performance degradation due to variations is benign if the timing is within the specification, and unlike chips with defects, they are safe to be used.

Figure 1 shows the produced delay bins for 4000 NAND cell instances in 14 nm FinFET technology only with variations (green bars) and those with an additional injected open defect (red bars). Already a large portion of the introduced delays have similar sizes, which makes it difficult to determine the source of the delay.

Figure 2 shows similar histograms for a circuit including a NAND cell in front of a chain of 16 inverters. The overlapped delay bins appear for a higher number of instances in larger circuits. In addition, variations from other cells in a circuit filter some important timing information of an embedded cell, which is necessary for defect identification. Fig. 3 shows the timing behavior of two defect-free NAND cells which are slow (dashed-green curve) and fast (solid

Responsible editor: L. M. B. Poehls

✉ Zahra Paria Najafi-Haghi
najafiza@informatik.uni-stuttgart.de

[1] Institute of Computer Architecture and Computer Engineering, University of Stuttgart, Pfaffenwaldring, Stuttgart 70569, Germany

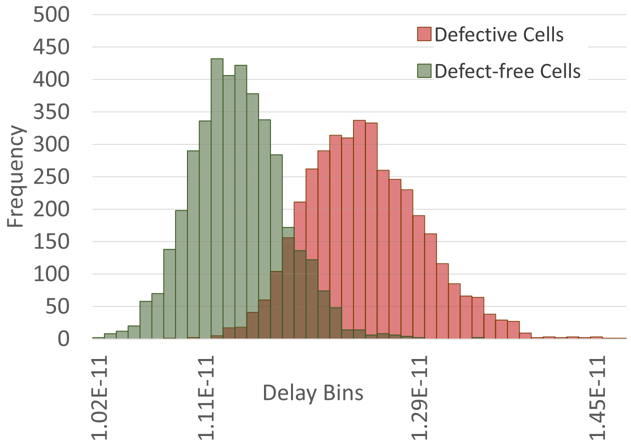


Fig. 1 Delay histograms of an isolated NAND cell at nominal voltage

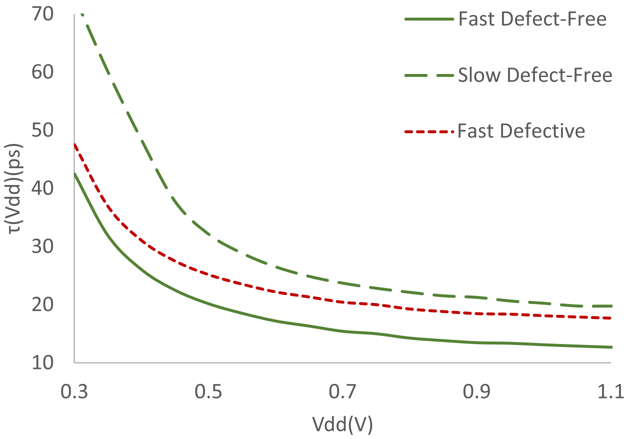


Fig. 3 Simulated delay vs. V_{dd} for a NAND Cell

green curve) due to variations. An instance with an injected open defect between NMOS transistors behaves as depicted by the dashed red curve. It is clear that a single measurement at any voltage cannot distinguish a defective cell from a defect-free one, since its timing is always between two defect-free timing and within the specification. The paper at hand enhances the method presented in [16] and analyzes the timing behavior of circuit instances, where an embedded cell is targeted under different operation voltages and applies statistical learning schemes to distinguish defective embedded cells from defect-free instances.

The main enhancements of this paper compared to [16] are as follows:

1. A complete defect identification approach is explained and formulated starting from isolated cells towards cells embedded in a circuit.

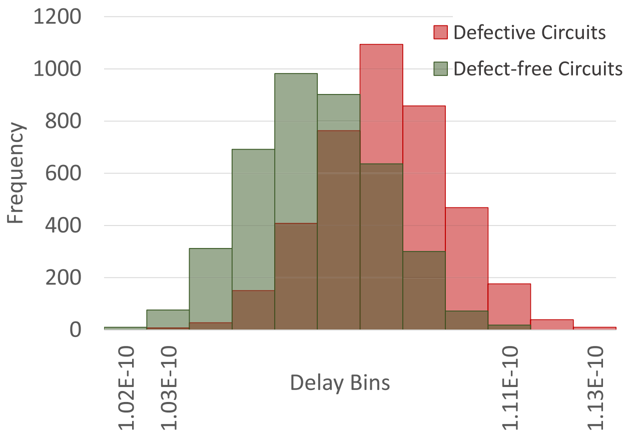


Fig. 2 Delay histograms of a NAND cell in front of a chain with 16 Inverters at nominal voltage

2. A precise variation-aware model is considered, in which the transistor models are updated [17] and the process-induced variability values and distributions are obtained from the recent industrial measurements by Intel [18, 19]. All experiments are performed from the scratch using the new real industrial data. Details will be further discussed in Section 2.
3. The applicability of the proposed approach is evaluated through additional combinational benchmark circuits.
4. The new realistic variability model leads to significantly improved results.

We show that filtering is transparent enough to distinguish the defect behavior from variations by machine learning techniques. Various statistical learning schemes are investigated and the one based on Random Forest (RF) can achieve an accuracy and a precision above 0.9 even for the largest circuit under investigation.

The analysis in this paper consists of the four major phases depicted in Fig. 4. The first two phases characterize the delay behavior of defect-free and defective circuits and result in the dataset required for the machine learning phase. The third phase selects and applies appropriate supervised machine learning-based classification schemes to the generated dataset, and finally, the created classifier is able to identify resistive open defects, even if the defect behavior is within the specification of the circuit.

The rest of the article is organized as follows. The next section gives some overview of the state of the art for defect identification under varying operating conditions. Section 3 describes some electronic fundamentals of delay faults under variations which will be exploited by the proposed strategy in this work. The challenges of defect identification under variations are explained in Section 4. The detailed steps of dataset generation by Monte Carlo Spice simulation, which is needed for training and validation of the machine

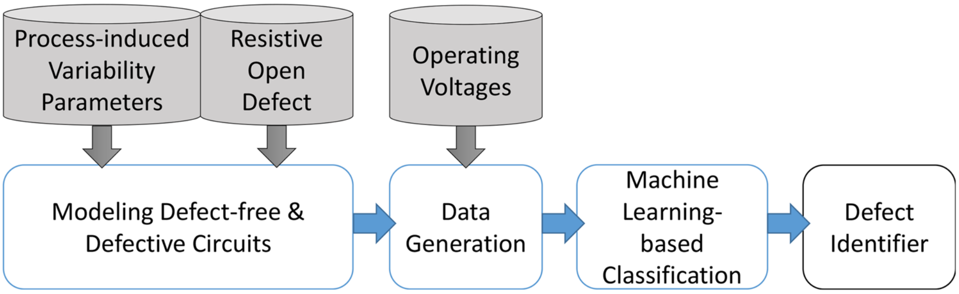

Process-induced Variability Parameters
Resistive Open Defect
Operating Voltages
Modeling Defect-free & Defective Circuits
Data Generation
Machine Learning-based Classification
Defect Identifier

Fig. 4 Creating an identifier for resistive open defects

learning-based approach are described in Section 5. The supervised machine learning-based classification techniques, as well as the validation metrics are presented in Section 6. Section 7 evaluates the performance of the proposed classification approach for different learning schemes and various cases from individual cells to embedded cells in circuits with different sizes. Remarks on further work and applications conclude the article in Section 8.

2 State of the Art

Physical weak defects such as resistive open, resistive bridge, and gate-oxide pinhole are considered as important sources of Early Life Failure (ELF), which appear as delay faults [20]. Delay test has been used to detect physical weak defects in a chip, e.g., [5, 10, 12, 15, 21–23].

Delay fault testing and the impact of operating conditions such as supply voltage have been investigated thoroughly in the past. It has been shown that reducing the supply voltage increases the transistor channel resistance, which results in an increasing electrical impact of a gate-drain (or gate-source) resistive bridge defect, or a drain (or source) resistive open defect [10]. Circuit test under varying operating conditions has been studied in many works such as [24, 25], which investigate the effect of supply voltage on the circuit delay and delay testing. The very-Low-Voltage test is exploited to detect ELF-related defects in [7, 8]. They studied the voltage dependence of CMOS logic circuit operation in the presence of physical flaws and showed that weak CMOS logic ICs can be forced to malfunction while truly good ICs continue to function at a certain much-lower-than-normal power supply voltage. The varying delay defect behavior at different voltages has been exploited in other works as well such as [26], which proposed the employment of testing at more than one supply voltage setting to improve defect coverage.

Since process-induced variations also produce timing variation in a chip, it needs to be considered in delay test [27], which is discussed in the concept of *variation-aware test* in [28]. The authors of [29] introduced a delay test method based on observing the outputs at multiple time intervals to detect the SDFs in the presence of variations.

As mentioned before, process-induced variation should be distinguished from physical weak defects [13] uses output delay correlation of two logic paths, which is called *inter-paths* information to screen out SDFs and distinguish them from process variations. It exploits the fact that delay measurements for a pair of paths must agree with their inter-path correlation. Otherwise, a defect is present in one of the paths. Delay test under varying voltages is exploited for the aim of distinguishment in [14, 21]. They use the fact that the relative delay contribution of an outlier transistor due to variations increases with decreasing V_{dd}, while the relative delay contribution of a resistive defect decreases with increasing V_{dd}. However, the investigations have been performed on conventional planar technology and the defect models lack precision in considering the variation-aware concept.

Machine learning-based classification to distinguish open defects from process-induced variations considering a more precise variation-aware defect model in the leading edge FinFET technology has been investigated in [16, 30, 31, 32] for detection of weak defects in cells, interconnects, embedded cells, and long paths of bigger circuits, respectively. However, the process-induced variability parameters in [16, 30, 31] consider a simplified model, which is sufficient for the proof of concept, but is not based on realistic measurements.

The paper at hand enhances the method presented in [16] and investigates the classification by using various machine learning techniques considering a precise variation-aware defect model.

Root causes of process-induced variability are metal-gate-granularity (MGG), line-edge-roughness (LER), which consists of fin-edge-roughness (FER) and gate-edge-roughness (GER) [19, 33], less impact have random-dopant-fluctuations (RDF) and oxide-thickness-variation (OTV). MGG determines the metal-work-function (MWF), and has an impact on both the sub-threshold swing as well as the threshold voltage, while LER mainly causes variations of the threshold voltage.

Unlike [16], which uses the simple free-PDK finFET transistor model, the paper at hand uses the industry-standard compact transistor model and process-induced variability parameters based on recent industrial measurements obtained from Intel for FinFET 14nm technology [18]. In particular, [16] considered transistor length (L) and width (W) as the variability parameters. The paper at hand models variations by more comprehensive parameters including transistor gate length (L), fin thickness (tfin), fin height (hfin), gate dielectric thickness (eot), and the impact of MGG on the work function of the gate (ϕ_g). The silicon-validated parameter distributions accurately model process variation in the transistor based on industrial measurements.

3 Electronic Analysis of Delay Faults under Variations

According to [34], the delay $\tau(V_{dd})$ of a CMOS transistor can be roughly expressed by the following equation.

$$\tau(V_{dd}) = \frac{C_L V_{dd} L T_{ox}}{\mu \epsilon_{ox} W (V_{dd} - V_t)^{\alpha}} \tag{1}$$

Here τ denotes the delay, C_L is the capacitance at the gate output, V_{dd} is the supply voltage, V_t is the transistor threshold voltage, α is the Sakurais index which can be taken equal to 1 in scaled technologies, L is the transistor channel length, W is the transistor channel width, μ is the carriers mobility, t_{ox} is the gate oxide thickness, and ϵ_{ox} is the gate oxide permittivity. For a robust analysis, other environmental conditions such as temperature are considered to be constant.

The delay $\tau(V_{dd})$ under various voltages (V_{dd}) for two defect-free NAND instances, which are fast and slow due to variations, can be seen as the solid and dashed green curves respectively in the plot of Fig. 3.

Threshold voltage fluctuation is considered as the major source of variations in timing [35] and is the focus of this work. Assume, V_t is now increased by δ due to process variations. By replacing V_t with $V_t + \delta$ and some mathematical reformulations, Eq. (1) is transformed into the new delay function $\tau'(V_{dd})$ as presented in Eq. (2).

$$\begin{aligned}\tau'(V_{dd}) &= \frac{C_L \times (V_{dd}) L T_{ox}}{\mu \epsilon_{ox} W \times (V_{dd} - (V_t + \delta))^{\alpha}} \\ &= \frac{C_L \times (V_{dd} - \delta) L T_{ox}}{\mu \epsilon_{ox} W \times (V_{dd} - \delta - V_t)^{\alpha}} + \frac{C_L \delta L T_{ox}}{\mu \epsilon_{ox} W \times (V_{dd} - \delta - V_t)^{\alpha}} \\ &= \tau(V_{dd} - \delta) + \tau(V_{dd} - \delta) \times \frac{\delta}{V_{dd} - \delta} \\ &= \tau(V_{dd} - \delta) \times \frac{V_{dd}}{V_{dd} - \delta} \\ & \qquad I \qquad\qquad II\end{aligned} \tag{2}$$

Part I of Eq. (2) shifts the solid green curve to the right by δ, and part II scales it decreasing with V_{dd}. Combining this results in the dashed green curve in Fig. 3, which is in fact observed in simulation.

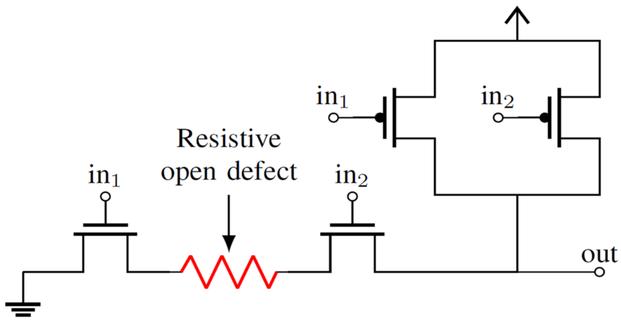


Fig. 5 A NAND-gate suffering from a resistive open defect

On the other hand, for constant V_t, the delay of a primitive gate can be expressed by the usual RC model Eq. (3).

$$\tau = C_L * (R_{eff}) \tag{3}$$

A resistive open will just increase R_{eff} by some δ'. The new delay is:

$$\tau' = C_L \times (R_{eff} + \delta') \tag{4}$$

which means a resistive defect of size δ' will shift the solid green curve of Fig. 3 upwards by $C_L * \delta'$ resulting in the dotted red curve. The simulated dotted red curve is a fast instance in which a resistive open is injected at the pull-down network, as shown in Fig. 5.

A similar analysis can be performed for resistive bridges and gate oxide pinholes, but is beyond the scope of this paper.

Figure 3 indicates that it may be impossible to distinguish a slow defect-free and a fast defective cell, based just on the delay measurement at a single voltage. However, the shapes of the two functions are sufficiently different, such that delay measurements at only 13 different voltages allow a highly accurate defect classification by statistical learning methods as seen in Section 7.

Each of the curves in Fig. 3 represents a delay $\tau(V_{dd})$, where the timing simulation is performed for a set V_{op} of voltages $V_{dd} \in V_{op}$. The vector

$$M^c = (\tau^c(V_{dd}) \mid V_{dd} \in V_{op}) \tag{5}$$

describes the measured performances for a produced chip $c \in C$ depending on the supply voltages. Each c has specific variability parameters. The vectors M^c for defect-free and defective chips are used further to train a machine learning-based classification technique, which is able to identify a defective new sample by its $\tau^c(V_{dd})$ vector.

In this work, we use transient analysis with SPICE to simulate the M^c vectors. In silicon production, these data can be obtained by real measurements as well.

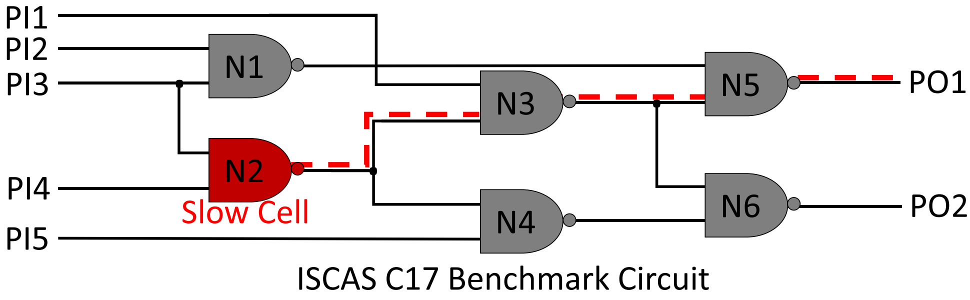


Fig. 6 An embedded slow cell in a synthetic combinational benchmark circuit

4 Masking Mechanisms and Defect Identification

It has been discussed before that the shapes of the two functions $\tau(V_{dd})$ for a defect-free and a defective cell may be sufficiently different to allow defect identification. However, the deeper a cell is embedded into combinational logic, the more its timing behavior will be blurred or even masked at the primary outputs.

Figure 6 shows an embedded cell under investigation, which can be subject to several masking effects described below.

- **Logical masking**: To detect a delay, a propagation path has to be sensitized. The discussion below assumes that only appropriate test patterns are used. ATPG is not the subject of this article.
- **Electrical masking**: CMOS is a self-restoring technology that filters short pulses and reshapes the slopes of transitions. Therefore, not only the cell under investigation but also the entire propagation path has to be subject to analog simulation. We will model the electrical masking by using an inverter chain, and analog simulation at a small combinational circuit. A section of the inverter netlist is shown in Fig. 8 where also the location of the resistive open defect is marked. The output of the embedded NAND cell is connected to a chain of λ inverters and all transistors are subject to individual random variations following a Gaussian distribution $N(\mu, \sigma)$. Figure 2 depicts a histogram of NAND cell with a resistive defect of the size of $3\sigma_\tau$ and a defect-free one, both in front of a chain of 16 inverters where all the transistors individually are subject to variations according to silicon-validated parameter distributions [19]. The points where the red and green curves of Fig. 3 get close are reflected here as those delay bins where red and green bins overlap. Exactly these bins form the challenge for any classification technique to be addressed in the next two sections.
- **Timing masking**: All the cells on the propagation path suffer from variations, and in general the propagation time of a path has to be modeled by a skewed multi-variable distribution [36, 37]. Since this distribution does not affect the output shape with respect to different voltages, it is assumed below that the path propagation delay follows a Gaussian distribution. The signal under observation has to be captured at the circuit output within a certain time window. Large defect sizes, which lead the circuit out of the specification will be detected by a standard delay test and are not the subject of our investigations. Small delay sizes, which do not affect the critical path, cannot be observed by any means. This defines a small interval of possible defect sizes, which intensifies the masking impact. If the defective cell is embedded in a combinational circuit with some convergences, all the three masking effects challenge the defect identification. The small benchmark circuit $C17$ from the ISCAS benchmark set [38] is used as an illustrative example (Fig. 6). The corresponding histogram considering various defect sizes from the smallest 0.5σ to the largest within the specification of the circuit (3σ) are shown in Fig. 7. Here σ is defined based on the delay distribution of the defective embedded cell (τ_{cell}). More details about the defect sizes are described in the following chapters. As expected, with the smaller defect size the overlapped delay range between defect-free and defective circuits is larger, and hence the defect identification is more challenging.

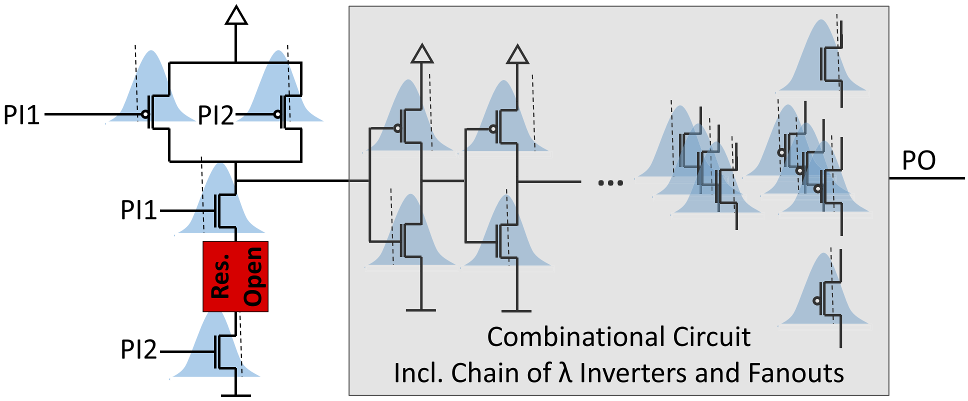
5 Data Generation

For a supervised learning scheme, the delay characterizations $M = \{M^c \mid c \in C\}$ of both defect-free and defective circuits have to be generated. The subset $DS_{df} \subset M$ gets the label *defect-free*, and $DS_d = M \setminus DS_{df}$ are the characterizations of defective circuits and receive the corresponding *defect* label. The details of the delay characterization are described below first for a single cell, and then for complete circuits.

5.1 Delay Characterization of Single Cells

Cell characterization is the process of modeling and measuring the characteristics of a single cell, e.g., propagation

Fig. 7 Delay histograms of C17 combinational benchmark circuit at the nominal voltage for various defect sizes within the specification of the circuit. (DF: Defect-free Circuits, D: Defective Circuits)



delays, pin transition times, power consumption, and setup/hold constraints. The resulting models are collected in a cell library to capture the delay measurements under a certain operation corner, in our case, a certain V_{dd}. This section describes how the required M^c vectors are simulated for both defect-free and defective cells from the open cell library (OCL).

5.1.1 Defect-free Cells

The machine learning procedures presented in Section 6 need simulation results of cell instances which follow the variability parameters of the underlying transistors. Hence, the standard formats for describing cell variability as the Library Variation Format (LVF) cannot be used, and a large standard cell library is built, which consists of hundreds of cells of the same cell type, only with different process-induced variability parameters.

The timing behavior of each of the cells is determined by SPICE simulation, with the already introduced model from [17].

All the mentioned variability parameters are modeled by a Gaussian distribution $N(\mu, \sigma)$. The nominal values μ are as defined in the industry-standard compact BSIM-CMG transistor model [17]. The process to assign the standard deviation values σ is explained in [19], in which the authors calibrated the values against industrial 14nm FinFET technology measurements from [18]. As stated in [19], $\sigma\phi_g/\mu\phi_g\% = 0.34\%$ with the $\mu\phi_g = 4.425V$ for the nFinFET transistors and $\mu\phi_g = 4.7V$ for the pFinFET transistors. The $\sigma/\mu\%$ for the remaining variability sources (L, $tfin$, $hfin$, eot) are set to 0.642%.

For supervised learning, a set TC of defect-free cells and a set FC of defective cells have to be generated. The propagation delay of each cell instance is characterized for a set of $V_{dd} \in V_{op}$ and stored as defect-free delay vectors into the corresponding dataset (DS_{df}).

$$DS_{df} = \{M^c \mid c \in TC\} \tag{6}$$

5.1.2 Defective Cells

To allow a balanced supervised learning scheme, a set FC of defective cell instances has to be generated, which has a similar amount as the defect-free one TC. First, another set of defect-free instances is generated as described before, then a weak resistive open defect is injected into each of the cells. Similar to the defect-free characterization, from each instance $c \in FC$ the same variation parameters are evaluated iteratively for different voltages to generate the vector M^c for c.

In this work, a single resistive open defect is injected into the critical area of the cell, which has a relatively high probability of getting a cut. To determine the critical area of each

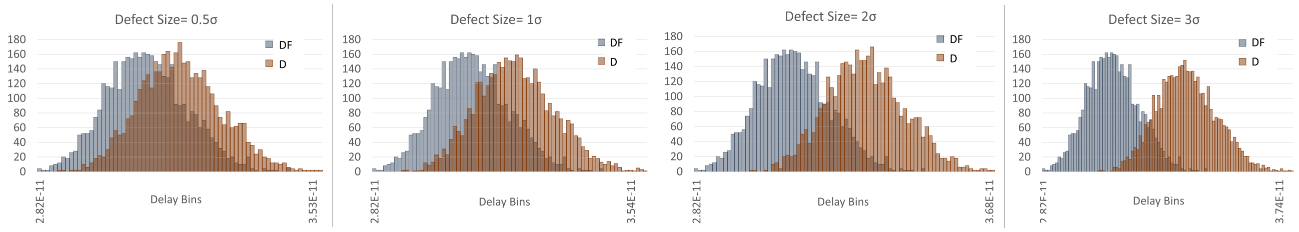


Fig. 8 Embedded NAND cell with the resistive open defect in front of an inverter chain

cell in the standard library, layout-based defect injection is performed as originally proposed as inductive fault analysis [39] and later commercialized as cell-aware test [40]. The defect sizes are quantized and def_{size} contains defects within the acceptable performance margin. A defective cell is within the specified margin of that cell if its delay under variations is less than $\mu_{\tau_{cell}} + 3\sigma_{\tau_{cell}}$, where $\mu_{\tau_{cell}}$ is the nominal delay of the NAND cell, and $\sigma_{\tau_{cell}}$ is the standard deviation due to variations. Clearly, any defect larger than this size is easier to identify. The resulting defective cell netlists ($FC(def_{size})$) capture the impact of the resistive open with the corresponding defect size in the cell.

By applying Monte-Carlo SPICE simulations, the defective netlists for each cell and each def_{size} are collected as a defective cell library. The propagation delay of each defective cell instance with a specific defect size def_{size} is then characterized for a set of $V_{dd} \in V_{op}$ and stored as defect delay vectors into the corresponding dataset for that defect size ($DS_{d(def_{size})}$).

$$\begin{aligned} DS_{d(def_{size})} &= \{M^c \mid c \in FC(def_{size})\} \\ &\quad def_{size} \in Def_{size} \end{aligned} \tag{7}$$

5.2 Chain of Inverters

In the previous subsection, we described the training and test dataset generation for an isolated cell. In order to investigate the electrical masking impact on the circuit delay ($\tau^c(V_{dd})$), we model a simplified combinational circuit including an embedded NAND cell in front of an inverter chain.

The inverter-chain is selected since it has the minimum number of transistors and often the highest impact of the variations on the delay behavior of the embedded cell. The parasitic elements and buffers are added to the chain to consider the fanout impact as well.

Figure 8 shows the location of the resistive open defect based on cell-aware analysis. A resistive open defect at a gate input would have a somewhat higher probability. The experimental results in Section 7 cover both cases. The output of the embedded NAND cell is connected to a chain of λ inverters. All transistors of the NAND cell and the inverter chain are subject to individual random variations following a Gaussian distribution $N(\mu, \sigma)$. More complex modeling with chains of more complex cells and even with individual and correlated distributions is possible [36, 37], but does not affect the arguments below. With this modeling, we do not need the complete layout of the circuit in applying the random variations for transistors. Correlated variations form often an easier case for classification, the exact analysis is left for further investigations.

The embedded NAND cell in the circuit is implemented by either a defect-free or a defective NAND to model the defect-free (TC) and defective circuits (FC). Monte-Carlo SPICE simulations are performed on TC and FC while varying the variability parameters to create the defect-free dataset (DS_{df}), and defect dataset (DS_d) by generating the M^c vectors in Eqs. (6) and (7). Various chain lengths λ are considered to investigate the filtering impact of the inverters on the delay propagation of the embedded cell.

If $\mu_{\tau_{emb}}$ is the nominal delay of the embedded NAND cell and $\mu_{\tau_{Inv}}$ is the nominal delay of each inverter, then Eq. (8) defines the expected delay of the entire circuit.

$$\mu_{\tau_{out}} = \mu_{\tau_{emb}} + \lambda * \mu_{\tau_{Inv}} \tag{8}$$

For the corresponding standard deviations $\sigma_{\tau_{emb}}$ and $\sigma_{\tau_{Inv}}$ of the cells, the standard deviation of the circuit output is as follows.

$$\sigma_{\tau_{out}} = \sqrt{\sigma^2_{\tau_{emb}} + \lambda * \sigma^2_{\tau_{Inv}}} \tag{9}$$

An inverter chain including a defective embedded NAND is considered within the specification if the circuit delay is less than $\mu_{\tau_{out}} + 3\sigma_{\tau_{out}}$

In summary, a large data set of modules will be generated for different chain lengths (λ), where all the transistors have individual random delays.

5.3 Benchmark Circuits

ISCAS C17 and ITC99 b02 benchmark circuits are selected as case studies to evaluate the proposed approach. The size of these circuits is selected so that their timing behavior could be analyzed by commercial and open-source SPICE simulators [41] in less than one day of computing time on a general-purpose processor. If a massive-parallel SPICE simulator for GPUs is available, e.g. [42], even circuits of several magnitudes larger can be validated.

5.3.1 ISCAS C17

The circuit C17 from ISCAS benchmark set [38] is used as an illustrative example in this work. Figure 6 shows the schematic of this circuit. C17 has 5 primary inputs ($PI1 - PI5$), 2 primary output ($PO1$, $PO2$), and 6 NAND gates ($N1 - N6$). In this experiment, the deepest gate, which is the $N2$ NAND gate, is considered as the slow embedded cell, and all transistors in the circuit are subject to variations. $N2$ is located in the output-cone for both POs, but we only analyze the behavior of output $PO1$. Observing more outputs will even improve the results further. The M^c vectors for defect-free and defective embedded NAND $N2$ are collected from $PO1$ and stored in datasets DS to train and validate the machine learning-based classifiers.

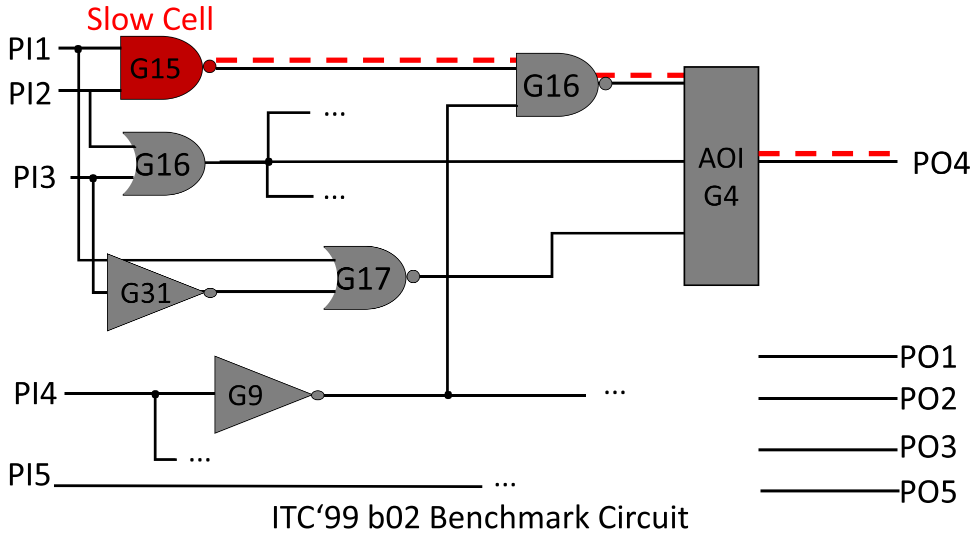


Fig. 9 A section of the b02 benchmark circuit including the slow NAND cell and its sensitized path to $PO4$

The embedded NAND $N2$ is either a defect-free NAND cell or a defective one, which is modeled as discussed in Section 5.1. The rest of the cells in the circuit suffer also from variations, which may cause timing and electrical masking. To build the dataset DS, Monte-Carlo SPICE simulations together with varying parameters are performed on the defect-free circuit instances TC as well as the circuit instances FC, in which the embedded NAND cell is replaced by a defective NAND. The M^c vectors are obtained from the PO1, and are divided into the defect-free (DS_{df}), and defect (DS_d) datasets according to Eqs. (6) and (7), to comprehensively investigate the quality of the defect identification.

5.3.2 ITC'99 b02

The circuit b02 from ITC99 benchmark set [43] is a Finite-State Machine (FSM) that recognizes Binary Coded Decimal (BCD) numbers, and the combinational version (-c) has 5 primary inputs ($PI1 - PI5$), 4 primary output ($PO1 - PO4$), and 21 standard gates (G1-G21). In this experiment, an embedded NAND gate (G15) is considered as the slow embedded cell, and all transistors in the circuit are subject to variations. Delay characterization of G15 can be observed from $PO4$. The M^c vectors for defect-free and defective embedded $G15$ NAND cells are collected from $PO4$ and stored in the dataset. Defect-free (TC) and defective (FC) circuit instances are modelled similar to $C17$, and the corresponding M^c vectors are stored according to Eqs. (6) and (7) in defect-free (DS_{df}), and defect (DS_d) datasets. The merged DS_{df} and DS_d builds the final dataset, which is used to train and validate the machine learning-based classifiers.

Figure 9 shows a section of the benchmark circuit, which includes the victim NAND cell and the corresponding $PO4$.

6 Classification by Supervised Learning

The dataset for the machine learning-based classification contains the M^c vectors corresponding to Eq. (5). The process to collect the delay vectors of defect-free circuit instances (TC) and defective ones (FC) is described in Sections 4 and 5 (Eqs. 6 and 7). DS_{df} and DS_d are combined to create the final dataset, which is used for the classifier. We use a supervised machine learning classifier, which means all samples in the training set (each row) should incorporate a label. Instances of the set DS_{df} get the label *defect-free* and the ones in the set DS_d come with the label *defective* in the dataset. The columns (features) are the supply voltages $V_{dd} \in V_{op}$, at which the delays are measured. Here the set V_{op} has ω members, i.e., supply voltages, and other controllable parameters such as temperature are constant in the created model. In this paper, voltage and frequency serve as features, since they are parameters rather easy to control and observe. Adding other parameters like temperature and current might improve the performance of the classification but would require more effort for implementing the measurements. They are outside the scope of this paper and left to further research.

6.1 Learning Schemes

The performance of three different statistical learning schemes is compared. The setup used for each scheme is described below.

- **Support Vector Machines (SVM)**: The Support Vector Machine algorithm separates data points with the largest margin into classes by constructing a hyper-plane between them. SVM is an effective supervised learning method in high dimensional spaces, and the kernel method is useful in implementing non-linear classification [44]. In this work, the Support Vector Classifier (SVC) with radial basis function (RBF) [45] kernel is used. RBF kernel is a kernel function, which is suitable for finding a non-linear classifier.

- *k* **-Nearest Neighbors (KNN)**: The *k*-Nearest Neighbors algorithm finds the distances of a new data point to the *k*-nearest data points and votes for the most frequent label. KNN algorithm is among the simplest and yet most efficient classification rules and is widely used in practice [46]. The distance is computed by the Euclidean metric.
- **Random Forest (RF)**: The Random Forest scheme is an ensemble learning scheme, which deploys bootstrap aggregating (bagging) of multiple decision trees, and evaluates the final result by majority voting. The number of trees is determined by the saturation of the classification accuracy. It may undertake dimensionality reduction methods to handle datasets with higher dimensionality [47]. A decision tree is a tree whose internal nodes can be taken as tests on input data patterns and whose leaf nodes can be taken as categories of these patterns [48]. Tree-based algorithms empower predictive models and map well to non-linear relationships as well as imbalanced data sets. Moreover, they do not get influenced by outliers to a fair degree and have high execution speed [49]. The Gini coefficient is used as the decision criteria, and the maximum depth of the tree is set to avoid over-fitting.

6.2 Evaluation

The classification quality for the three schemes introduced before is evaluated with respect to the standard metrics used in statistical learning [50]. The attribute P of an instance means defect (positive), and N means no-defect (negative). A correct classification is true (T), otherwise false (F). The results of the test instances are partitioned into four groups.

- True Positive (TP): Defective instances correctly identified as defective.
- True Negative (TN): Defect-free instances correctly identified as defect-free.
- False Positive (FP): Defect-free instances wrongly classified as defective.
- False Negative (FN): Overlooked defective instances wrongly classified as defect-free.

Precision Precision indicates the ratio of the correctly classified instances, over all the classified instances of that class. Usually, different values for the class defect (D) and no-defect (ND) are observed.

$$Prec_D := \frac{|TP|}{|TP| + |FP|}, Prec_{ND} := \frac{|TN|}{|TN| + |FN|} \tag{10}$$

$1 - Prec_D$ gives us the ratio of false alarms from all the alarms.

Recall represents the ratio of the correctly classified instances in a class, over the entire class instances. Similar to precision, different values for the class defect (D) and no-defect (ND) are observed.

$$Recall_D := \frac{|TP|}{|TP| + |FN|}, Recall_{ND} := \frac{|TN|}{|TN| + |FP|} \tag{11}$$

$Recall_D$ can be considered as the defect coverage, and $1 - Recall_{ND}$ relates to the unwanted yield loss.

Accuracy denotes how much of the test data are in total classified correctly. Accuracy is prone to get biased by imbalanced data sets. Assuming N' as the size of the overlap test data, accuracy is defined as follows.

$$Accuracy := \frac{|TP| + |TN|}{N'} \tag{12}$$

The **F1-score** is defined as the harmonic mean of the corresponding precision and recall for the defect (D) and no-defect (ND).

$$F1 - score_{(D,ND)} := \frac{2}{1/Recall_{(D,ND)} + 1/Precision_{(D,ND)}} \tag{13}$$

All the metrics mentioned above are applied to the three Machine Learning (ML) schemes. To generate a robust and fair evaluation, K-fold cross-validation is used [51]. The data set of N' instances is partitioned into $k = 10$ disjoint randomly selected subsets. In a round-robin fashion, the label is removed from one subset in order to be used as a test set, and the union of the remaining sets is used for training. The metrics above are obtained as the average outcome of these $k = 10$ experiments.

7 Simulation Results

To evaluate the quality of the trained classifier, several experiments have been performed. In the first subsection, the three ML schemes are compared according to the different metrics mentioned in the previous section for a single NAND cell from OCL. The second and third subsections present the evaluation results for an embedded cell in an inverter chain with various lengths, and the b02 circuit from the ITC99 benchmark respectively.

The set V_{op} of supply voltages is defined by the interval $[0.4, \dots, 1.0]$ with a step size of 0.05 and requires 13 measurements for $F^c_{max}(V_{dd})$, for $V_{dd} \in V_{op}$. The number of generated instances (n) for each defect-free and defect dataset (DS_{df} and DS_d) is 2000.

Table 1 Classification results for an individual NAND cell

Schemes	Defect Size	$Prec_D$	$Prec_{ND}$	$Recall_D$	$Recall_{ND}$	$F1-score_D$	$F1-score_{ND}$	$Accuracy$
kNN	$1\sigma, 2\sigma, 3\sigma$	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	0.5σ	1.0	0.998	0.999	1.0	1.0	0.999	0.999
SVM	$1\sigma, 2\sigma, 3\sigma$	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	0.5σ	1.0	0.984	0.989	1.0	0.996	0.993	0.993
RF	$1\sigma, 2\sigma, 3\sigma$	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	0.5σ	1.0	0.990	0.995	0.997	0.997	0.994	0.994

7.1 Defect identification for an isolated NAND cell

As described in Subsection 5.1.2, the two most critical defect locations are extracted from cell-aware test and a single resistive open defect with four various defect sizes of $3\sigma_{\tau_{emb}}$, $2\sigma_{\tau_{emb}}$, and $1\sigma_{\tau_{emb}}$, and even $0.5\sigma_{\tau_{emb}}$ is injected once at a time such that the new delay of the defective instance (τ') with an additional delay due to defect satisfies Eq. (14). One defect location is as shown in Fig. 5, and the second one is on the primary output.

$$\tau'_{i\sigma} = \mu_{\tau_{cell}} + i\sigma_{\tau_{cell}} \tag{14}$$

for $i = 0.5, 1, 2, 3$

The delay characterization of defect-free and defective single NAND cells are simulated and stored in DS_{df} and DS_d respectively as described in Subsections 5.1. All three machine learning-based classification schemes are applied on the generated datasets, and the classification results are evaluated based on K-fold cross-validation, where K equals 10. The numbers TP, TN, FN, and FP are extracted for the dataset of each fold and the average of the evaluation metrics are reported for each classifier using Scikit-learn [52]. The classification metrics for an individual NAND cell can be seen in Table 1.

The classification results reported in Table 1 means for all three larger defect sizes of 1σ, 2σ, and 3σ, all instances can be classified correctly using any of the three machine learning-based classification schemes. We reduce the defect size to even 0.5σ, but still, all schemes can classify defect-free and defective instances with very high precision, recall, and accuracy between 99 to 100 percent. It should be noted that all these four defect sizes lead to a behavior within the specification of the cell. For the rest of this paper, only two smaller defect sizes of 1σ and 0.5σ, which are harder to be classified are presented.

7.2 Defect Identification for an Embedded Cell in an Inverter Chain

Defect-free and defective NAND cells are embedded in the front of a chain of inverters to analyze the masking impact on its timing information in combinational circuits. The chain has various lengths (λ) of 2, 4, 8, and 16 to investigate the impact of the sensitized path depth on the defect identification of an embedded cell.

The delay characterization of defect-free and defective circuit instances are simulated by Monte Carlo SPICE simulation and stored in DS_{df} and DS_d respectively as described in Subsections 5.2.

Tables 2 and 3 present the evaluation metrics for the developed classifier based on each machine learning scheme for the 1σ and 0.5σ defect size from equations Eq. (14) respectively. The most left column denotes the number of inverters in the chain (λ) or in other words the chain length.

The impact of the defect size from Eq. (14) on the classification quality can be seen by comparing Tables 2 and 3, in which the embedded NAND cell is injected with a 1σ and 0.5σ defect size respectively. The general trend is that with increasing defect size the defect-free and defective instances behave more distinctly and can be easier classified. Therefore, the performance of all schemes is increasing further. For all larger defect sizes within the specification, we observed around 100% classification quality, which for the sake of brevity is not presented here. The impact of the circuit depth, which is the main cause of the timing information masking in a circuit can be observed for each fixed defect size, by moving from the up to the downside of the table. The classification results reported in both Tables 2 and 3 show a constant decrease in the classification quality by increasing the circuit depth. This is due to the fact that in larger circuits, more cells appear on the sensitized path from the slow embedded cell to the observation point on the primary output. Each cell on the path has an additional masking impact on the timing information of the embedded cell, which results in filtering some useful timing information from the embedded cell and makes the classification harder. On the other hand, Eqs. (8) and (9) show how the distribution of the circuit delays will change and gets wider deviation by having more cells in the circuit. This results in having a larger overlap between delay characterization of defect-free and defective circuits, which makes the classification even more challenging.

7.3 Defect Identification for an Embedded Cell in C17 and B02 Benchmark Circuits

To investigate the additional masking impact of fan-out and multiple paths on the defect classification of embedded cells, two combinational benchmark circuits C17, and b02 from ISCAS and ITC99 benchmark sets are investigated.

Table 2 Classification results for an embedded NAND cell with defect size of 1σ in an inverter chain

λ	Schemes	$Prec_D$	$Prec_{ND}$	$Recall_D$	$Recall_{ND}$	$F1-score_D$	$F1-score_{ND}$	$Accuracy$
2	kNN	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	SVM	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	RF	1.0	1.0	1.0	1.0	1.0	1.0	1.0
4	kNN	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	SVM	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	RF	1.0	1.0	1.0	1.0	1.0	1.0	1.0
8	kNN	0.995	0.98	0.973	0.998	0.984	0.988	0.985
	SVM	0.994	0.972	0.962	0.998	0.976	0.984	0.981
	RF	0.998	0.982	0.978	0.999	0.988	0.991	0.989
16	kNN	0.975	0.952	0.939	0.98	0.957	0.964	0.961
	SVM	0.979	0.95	0.936	0.983	0.958	0.967	0.96
	RF	0.996	0.972	0.964	0.999	0.98	0.985	0.982

Defect-free and defective embedded NAND cells are placed in each circuit and its timing information is collected from a primary output as described in Subsections 5.3.1 and 5.3.2. The delay characterization of defect-free and defective circuit instances are simulated by Monte Carlo SPICE simulation and stored in DS_{df} and DS_d respectively.

Tables 4 and 5 present the evaluation metrics for the developed classifier based on each machine learning scheme for the $C17$ and $b02$ circuit respectively. The most left column in each table denotes the defect size considered in the simulation of each circuit, which here are the two most challenging defect sizes of 1σ and 0.5σ from Eq. (14).

For $C17$, which is a smaller benchmark circuit, the defect size of 1σ can be classified with 100% accuracy, and the smaller defect size of 0.5σ can also be classified with all the metrics above 96%. The lowest metric, which belongs to $Prec_{ND}$ by using SVM means 96% of all the instances, which are classified as defect-free are correctly classified and in only 4% a warning should have been given as defective. The similar metric for defect classification ($Prec_D$) in the same scenario, shows 99%, which means 99% of the instances, which are classified as defective, are in fact defective and only 1% is a false alarm. Depending on the yield, this metric can also be used as reliability binning. Here, it should be considered that the experiment is performed with a balanced data set. In practice, the defective parts should be much less than 50%, and the portion of overlooked cases will be much less as well.

In $b02$, which is larger than $C17$, and has more reconvergences from other paths to the sensitized path of the slow embedded cell, we observe a lower classification quality. But still, all the schemes reach an accuracy significantly larger than 98%, and all other metrics are above 96%. The lowest $Recall_D$ belongs to 0.5σ when using the kNN scheme. $Recall_D = 0.979$ means that only 2% of marginal chips will pass, and $Recall_{ND} = 0.998$ when using RF denotes a yield loss of only 0.2% by sorting out correct chips.

It is observed that for almost all circuits and scenarios the best results are obtained by the RF scheme. This is due to the fact that tree-based classification schemes map well to non-linear relationships, as well as imbalanced data sets,

Table 3 Classification results for an embedded NAND cell with defect size of 0.5σ in an inverter chain

λ	Schemes	$Prec_D$	$Prec_{ND}$	$Recall_D$	$Recall_{ND}$	$F1-score_D$	$F1-score_{ND}$	$Accuracy$
2	kNN	0.988	0.956	0.942	0.993	0.964	0.972	0.969
	SVM	0.987	0.941	0.924	0.994	0.956	0.964	0.960
	RF	0.980	0.957	0.941	0.986	0.962	0.971	0.965
4	kNN	0.987	0.950	0.958	0.984	0.973	0.967	0.969
	SVM	0.979	0.947	0.952	0.975	0.968	0.960	0.961
	RF	0.979	0.950	0.956	0.972	0.968	0.962	0.964
8	kNN	0.957	0.939	0.920	0.966	0.938	0.951	0.945
	SVM	0.958	0.917	0.896	0.967	0.926	0.943	0.936
	RF	0.960	0.931	0.910	0.970	0.934	0.950	0.943
16	kNN	0.884	0.874	0.840	0.909	0.861	0.891	0.877
	SVM	0.910	0.785	0.697	0.942	0.787	0.857	0.828
	RF	0.924	0.893	0.861	0.940	0.889	0.918	0.905

Table 4 Classification results for an embedded NAND cell in $C17$ benchmark circuit

Defect Size	Schemes	$Prec_D$	$Prec_{ND}$	$Recall_D$	$Recall_{ND}$	$F1 - score_D$	$F1 - score_{ND}$	$Accuracy$
1σ	kNN	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	SVM	1.0	0.99	1.0	1.0	1.0	1.0	1.0
	RF	1.0	0.99	1.0	1.0	1.0	1.0	1.0
0.5σ	kNN	0.994	0.978	0.983	0.990	0.988	0.985	0.985
	SVM	0.995	0.962	0.969	0.990	0.981	0.976	0.978
	RF	0.995	0.980	0.987	0.992	0.990	0.988	0.988

and get less influenced by outliers. In addition, RF can boost the tree-based models by executing bias-variance trade-off analysis. It also selects the features based on classification scores, which enables it to handle a big set of data.

It has to be noted that only the N' hard-to-classify instances, which can be seen as the overlapped range in Figs. 1, 2 and 7 are under investigation. Without the proposed technique to distinguish defect from defect-free, this part of the produced chips (N') should be either completely removed from the final product which means a high and expensive yield loss, or be kept, which increases the rate of ELF.

The Monte Carlo simulation time to generate the M^c vectors for 2000 instances of each circuit is in the order of hours and the run-time for training and validation with machine learning-based classification in Python is in the order of seconds.

7.4 Application Scenarios

According to the presented results, the RF scheme is able to identify resistive opens with high classification quality. This result is encouraging enough to train such an ML model on real silicon data for which industrial support is needed. It has been shown that the electrical and timing masking are transparent enough to convey the useful timing information needed for an accurate defect classification for embedded cells in the presence of variations.

To investigate the proposed method on larger benchmark circuits, dataset generation with the accurate but slow HSPICE simulation is not feasible. Instead, a more comprehensive cell characterization under various conditions can be exploited to speed up the process. The exact steps for a new gate-level netlist are as follows:

- Characteristics of the standard cells are simulated by SPICE Monte Carlo and stored in an extended cell library, in which each standard cell has multiple instances and each instance has unique variability parameter values.
- The same amount of instances for each cell type are simulated with different variability parameter values and additionally injected with a defect to build the population of defective standard cell instances and stored in the extended cell library.
- The generated cell characterization library should be done once and can be sampled for the analysis of large circuits later.
- To reduce the complexity of the cell characterizations, methods such as [53] based on machine learning have been proposed, which speeds up the characterization process.
- Tools such as timing-aware simulation tool [54] as well as Static Timing Analysis (STA) [55] can randomly select from the already generated cell characterization library and use them for timing analysis and generating the required vectors M^c.
- During the chip manufacturing, vectors M^c can be obtained directly from real measurements at the speed-binning [32] or faster-than-at-speed test [4].

Table 5 Classification results for an embedded NAND cell in $b02$ benchmark circuit

Defect Size	Schemes	$Prec_D$	$Prec_{ND}$	$Recall_D$	$Recall_{ND}$	$F1 - score_D$	$F1 - score_{ND}$	$Accuracy$
1σ	kNN	1.0	0.996	0.992	1.0	0.998	0.997	0.997
	SVM	1.0	0.995	0.989	1.0	0.997	0.997	0.996
	RF	0.996	0.992	0.990	0.999	0.993	0.996	0.993
0.5σ	kNN	0.995	0.984	0.979	0.999	0.989	0.990	0.990
	SVM	1.0	0.969	1.0	0.962	0.981	0.983	0.982
	RF	0.992	0.984	0.979	0.998	0.987	0.990	0.988

8 Conclusion and Further Work

Resistive opens can be identified in a cell, even if it is deeply embedded into a combinational circuit and does not change the circuit behavior beyond the specification. A machine learning scheme based on Random Forest is able to classify these cells under variations with very high accuracy. The encouraging result can be used for quality screening, binning, and diagnosis with a negligible impact on the yield.

Funding Open Access funding enabled and organized by Projekt DEAL. This work is part of the project grant WU 245/19-1 (FAST) and WU 245/22-1 (ACCROSS) funded by the German Research Foundation (DFG), and is partly supported by Advantest as part of the Graduate School Intelligent Methods for Test and Reliability (GS-IMTR) at the University of Stuttgart.

Data Availability The datasets generated during and/or analyzed during the current study are available from the corresponding author upon reasonable request.

Declarations

Conflicts of Interest The authors declare that they have no conflict of interest or competing interests.

References

1. Forero F, Galliere J, Renovell M, Champac V (2018) Detectability Challenges of Bridge Defects in FinFET Based Logic Cells. J Electron Test 34(2):123–134
2. Czutro A, Houarche N, Engelke P, Polian I, Comte M, Renovell M, Becker B (2008) A Simulator of small-delay faults caused by resistive-open defects. In: Proc. IEEE European Test Symp (ETS) pp. 113–118
3. Engelke P, Polian I, Renovell M, Kundu S, Seshadri B, Becker B (2008) On detection of resistive bridging defects by low-temperature and low-voltage testing. IEEE Trans Comput-Aided Des Integr Circuits Syst 27(2):327–338
4. Hellebrand S, Indlekofer T, Kampmann M, Kochte M, Liu C, Wunderlich H (2014) FAST-BIST: Faster-than-at-Speed BIST Targeting Hidden Delay Defects. In: Proc IEEE Int'l Test Conf (ITC), pp. 1–8
5. Hapke F, Schloeffel J, Redemund W, Glowatz A, Rajski J, Reese M, Rearick J, Rivers J (2011) Cell-aware analysis for small-delay effects and production test results from different fault models. In: Proc IEEE Int'l Test Conf (ITC), pp. 1–8
6. Liu C, Schneider E, Kampmann M, Hellebrand S, Wunderlich H (2018) Extending aging monitors for early life and wear-out failure prevention. In: Proc. IEEE Asian Test Symp (ATS) pp. 92–97
7. Hao H, McCluskey EJ (1993) Very-Low-Voltage testing for weak cmos logic ICs. In: Proc IEEE Int'l Test Conf (ITC) pp. 275–284
8. Chang JT, McCluskey EJ (1996) Detecting delay flaws by very-low-voltage testing. In: Proc. IEEE Int'l Test Conf (ITC) pp. 367–376
9. Engelke P, Polian I, Renovell M, Seshadri B, Becker B (2004) The pros and cons of very-low-voltage testing: An analysis based on resistive bridging faults. In: Proc IEEE VLSI Test Symp (VTS) pp. 171–178
10. Qian X, Chao Han, Singh AD (2012) Detection of gate-oxide defects with timing tests at reduced power supply. In: Proc IEEE VLSI Test Symp (VTS) pp. 120–126
11. Mhira S, Huard V, Benhassain A, Cacho F, Naudet S, Jain A, Parthasarathy C, Bravaix A (2017) Dynamic adaptive voltage scaling in automotive environment. In: Proc IEEE Int'l Reliability Physics Symp (IRPS) pp. 341–347
12. Aitken RC (2008) Defect or Variation? characterizing standard cell behavior at 90 nm and below. IEEE Trans Semicond Manuf 21(1):46–54
13. García-Gervacio JL, Nocua A, Champac V (2015) Screening small-delay defects using inter-path correlation to reduce reliability risk. Microelectron Reliab 55(6):1005–1011
14. Qian X, Singh AD (2010) Distinguishing Resistive Small Delay Defects from Random Parameter Variations. In: Proc IEEE Asian Test Symp (ATS) pp. 325–330
15. Karel A, Azais F, Comte M, Galliere J, Renovell M, Singh K (2017) Detection of resistive open and short defects in fdsoi under delay-based test: optimal vdd and body biasing conditions. In: Proc IEEE European Test Symp (ETS) pp. 1–2
16. Najafi-Haghi ZP, Wunderlich HJ (2021) Resistive Open Defect Classification of Embedded Cells under Variations. In: Proc IEEE Latin American Test Symp (LATS) pp. 1–6
17. Khandelwal S, et al (2014) BSIM-CMG 110.0.0: Multi-gate MOSFET Compact Model: Technical Manual
18. Natarajan S, et al (2014) A 14nm Logic Technology Featuring 2nd-Generation FinFET, Air-Gapped Interconnects, Self-Aligned Double Patterning and a 0.0588 μm2 SRAM Cell Size. In: Proc. IEEE Intl Electron Devices Meeting. pp 371–373
19. Amrouch H, Pahwa G, Gaidhane AD, Dabhi CK, Klemme F, Prakash O, Chauhan YS (2020) Impact of variability on processor performance in negative capacitance finFET technology. IEEE Trans Circuits Syst 67(9):3127–3137
20. Forero F, Galliere J, Renovell M, Champac V (2017) Analysis of short defects in finfet based logic cells. In: Proc IEEE Latin American Test Symp (LATS) pp. 1–6
21. Haihua Y, Singh AD (2005) A Delay test to differentiate resistive interconnect faults from weak transistor defects. In: Proc. Int'l Conf of VLSI Design (VLSID) pp. 47–52
22. Khursheed S, Zhong S, Aitken R, Al-Hashimi BM, Kundu S (2010) Modeling the Impact of Process Variation on Resistive Bridge Defects. In: Proc IEEE Int'l Test Conf (ITC) pp. 1–10
23. Liu Y, Xu Q (2012) On Modeling faults in finfet logic circuits. In: Proc. IEEE Int'l Test Conf (ITC) pp. 1–9
24. Wagner KD, McCluskey EJ (1985) Effect of supply voltage on circuit propagation delay and test applications. In: Proc IEEE Int'l Conf on Computer-Aided Design (ICCAD) pp. 42–44
25. Zain Ali NB, Zwolinski M, Al-Hashimi BM, Harrod P (2006) Dynamic voltage scaling aware delay fault testing. In: Proc IEEE European Test Symp. (ETS), pp 15-20
26. Khursheed S, Ingelsson U, Rosinger P, Al-Hashimi BM, Harrod P (2008) Bridging fault test method with adaptive power management awareness. IEEE Trans Comput-Aided Des Integr Circuits Syst 27(6):1117–1127

27. Lu X, Li Z, Qiu W, Walker DMH, Shi W (2004) Longest Path Selection for Delay Test under Process Variation. In: Proc. IEEE Asia-South Pacific Design Automation Conf. (ASP-DAC) pp. 98–103
28. Sauer M, Polian I, Imhof ME, Mumtaz A, Schneider E, Czutro A, Wunderlich HJ, Becker B (2014) Variation-Aware Deterministic ATPG. In: Proc. IEEE European Test Symp. (ETS), pp 1-6
29. Yan H, Singh A (2004) On the Effectiveness of Detecting Small Delay Defects in the Slack Interval. In: Proc. IEEE Int'l Workshop on Current and Defect Based Testing, pp 49-53
30. Najafi-Haghi Z, Hashemipour Nazari M, Wunderlich H (2020) Variation-Aware Defect Characterization at Cell Level. In: Proc. IEEE European Test Symp. (ETS), pp 1-6
31. Sprenger A, Sadeghi-Kohan S, Reimer JD, Hellebrand S (2020) Variation-Aware Test for Logic Interconnects using Neural Networks-A Case Study. In: Proc. IEEE Int'l. Symp. on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), pp 1-6
32. Najafi-Haghi ZP, Klemme F, Amrouch H, Wunderlich HJ (2022) On Extracting Reliability Information from Speed Binning. In: Proc. IEEE European Test Symp. (ETS), pp 1-4
33. Wang X, Brown AR, Cheng B, Asenov A (2011) Statistical Variability and Reliability in Nanoscale FinFETs. In: Proc. Intl. Electron Devices Meeting. pp 541–544
34. Champac V, Gervacio JG (eds) (2018) Timing Performance of Nanometer Digital Circuits Under Process Variations. Springer, International Publishing
35. Yanambaka VP, Mohanty SP, Kougianos E, Ghai D, Ghai G (2017) Process Variation Analysis and Optimization of a FinFET-Based VCO. IEEE Trans Semicond Manuf 30(2):126-134
36. Wagner M, Wunderlich HJ (2013) Efficient Variation-Aware Statistical Dynamic Timing Analysis for Delay Test Applications. Proc. ACM/IEEE Conf. on Design, Automation Test in Europe (DATE), pp 276-281
37. Wagner M, Wunderlich HJ (2017) Probabilistic Sensitization Analysis for Variation-Aware Path Delay Fault Test Evaluation. In: Proc. IEEE European Test Symp. (ETS), pp 1-6
38. Brglez F, Fujiwara H (1985) A Neutral Netlist of 10 Combinational Benchmark Circuits and a Targeted Translator in FORTRAN. pp 677-692
39. Shen JP, Maly W, Ferguson FJ (1985) Inductive Fault Analysis of MOS Integrated Circuits. IEEE Design and Test 6:13-26
40. Maxwell P, Hapke F, Tang H (2016) Cell-Aware Diagnosis: Defective Inmates Exposed in their Cells. In: Proc. IEEE Int'l European Test Symp. (ETS), pp 1-6
41. Synopsys (2019) HSPICE. https://www.synopsys.com/content/dam/synopsys/verification/datasheets/hspice-ds.pdf. Accessed 3 Feb 2023
42. van Santen VM, Diep FLF, Henkel J, Amrouch H (2020) Massively Parallel Circuit Setup in GPU-SPICE. IEEE Trans on Computers (Early Access) pp 1-1
43. Davidson S (1999) ITC'99 Benchmark Circuits - Preliminary Results. In: Proc. of IEEE Int'l Test Conf. (ITC), pp 1125-1125
44. Gualtieri JA, Chettri S (2000) Support vector machines for classification of hyperspectral data. In: Proc. IEEE Int'l Geoscience and Remote Sensing Symp. (IGARSS), vol 2. pp 813–815
45. Schölkopf B, Smola AJ (ed) (2001) Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, London
46. Laaksonen J, Oja E (1996) Classification with Learning K-nearest Neighbors. In: Proc. of Int'l Conf. on Neural Networks (ICNN), vol 3, pp 1480-1483
47. Tin Kam Ho (1995) Random Decision Forests. In: Proc. of Int'l Conf. on Document Analysis and Recognition (ICDAR), pp 278-282
48. Navada A, Ansari AN, Patil S, Sonkamble BA (2011) Overview of Use of Decision Tree Algorithms in Machine Learning. In: IEEE Control and System Graduate Research Colloquium, pp 37-42
49. Quinlan JR (1993) C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco
50. Marsland S (2015) Machine Learning: An Algorithmic Perspective, 2nd edn. CRC Press, Florida
51. Kohavi R (1995) A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. In: Proc. Int'l Joint Conf. on Artificial Intelligence (IJCAI), pp 1137-1143
52. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: Machine learning in Python. J Mach Learn Res 12(Nov):2825–2830
53. Klemme F, Amrouch H (2022) Efficient Learning Strategies for machine learning-based aging-aware cell library characterization. IEEE Trans Circuits Syst 69(12):5233–5246
54. Schneider E, Wunderlich HJ (2020) GPU-Accelerated Time Simulation of Systems with Adaptive Voltage and Frequency Scaling. In: Proc. ACM/IEEE Conf. on Design, Automation Test in Europe (DATE)
55. Bhasker J, Chadha R (2009) Static Timing Analysis for Nanometer Designs: A Practical Approach. Springer Science & Business Media, New York

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Zahra Paria Najafi-Haghi is a student member of IEEE. She received the Bachelor's and Master's degrees in Computer Engineering and Computer Architecture from the University of Tehran, Iran. Since January 2017, she has started working toward Ph.D. at the institute of Computer Architecture and Computer Engineering, University of Stuttgart, Germany, under the supervision of Professor Hans-Joachim Wunderlich. Her research interests include Test and Reliability in Digital Circuits, Machine Learning for Test, Hidden Delay Fault Test, and Variation-aware Test.

Hans-Joachim Wunderlich is a Life Fellow of IEEE. He received the diploma degree in mathematics from the University of Freiburg, Germany, in 1981 and the Dr. rer. nat. (Ph.D. degree) from the University of Karlsruhe in 1986. Since 1991, he has been a full professor. From 2002 to 2018, he was the director of the Institute of Computer Architecture and Computer Engineering at the University of Stuttgart, Germany. He has been associate editor of various international journals and program committee member of a variety of IEEE conferences on design, test and fault tolerance of electronic systems. He has published 15 books and book chapters and around 300 reviewed scientific papersin journals and conferences. His research interests include test, reliability, fault tolerance and design automation of microelectronic systems.