American Academic Scientific Research Journal for Engineering, Technology, and Sciences

ISSN (Print) 2313-4410, ISSN (Online) 2313-4402

https://asrjetsjournal.org/index.php/American_Scientific_Journal/index

Novel Machine Learning Approach for Defect Detection in DFT Processes

Vijayaprabhuvel Rajavel*

Semiconductor Design & Test Engineering Specialist, California, USA

Email: vijayscholarly@gmail.com

Abstract

Recent advances in semiconductor technology have highlighted significant challenges in effectively testing modern integrated circuits (ICs). As device densities increase and defect mechanisms become more diverse, conventional Design for Testability (DFT) methodologies – while indispensable – must contend with exponential growth in test complexity. This paper reviews the essential DFT practices, including scan-based structures, boundary scan, and built-in self-test (BIST), and examines how these practices address a variety of logical fault models. It further explores machine learning (ML) techniques as valuable tools for enhancing defect detection and diagnosis. By leveraging classification algorithms such as support vector machines and neural networks, ML-driven approaches can reduce test pattern generation time, improve bridging-fault coverage, and streamline board- or wafer-level screening. Collectively, this paper underscores how strategic synergy between DFT and ML can raise fault coverage, improve diagnostic precision, and contain testing costs in the face of ongoing technology scaling.

Keywords: Design for Testability (DFT); Machine Learning (ML); Scan Testing; Bridging Faults; Open/Short Defects; Built-In Self-Test (BIST); Automatic Test Pattern Generation (ATPG); Semiconductor Yield.

1. Introduction

Design for Testability (DFT) encompasses a suite of architectural enhancements and methodologies – such as scan chains, built-in self-test (BIST), and boundary scan – that simplify the testing of integrated circuits (ICs) by improving internal observability and controllability while reducing the number of required test patterns [1, 2]. As semiconductor technologies advance into deep- and sub-nanometer nodes, modern SoCs now integrate billions of transistors, rendering conventional testing approaches (e.g., deterministic automatic test pattern generation, ATPG) increasingly intractable due to near-exponential growth in pattern-generation complexity and fault-diagnosis overhead [3].

Received: 2/27/2025
Accepted: 3/28/2025
Published: 4/22/2025

----

* Corresponding author.

Driven by Moore's Law [4], device scaling has also introduced a proliferation of subtle defect modes: besides classic stuck-at faults, advanced processes exhibit open/short bridging failures, parametric drifts, and intermittent anomalies that can evade standard test vectors [5]. While augmented DFT techniques – such as boundary scan at the board level or in-field BIST – capture a broader fault spectrum, they add significant area and timing overhead, may obscure complex failure signatures, and still rely on robust diagnostic algorithms to resolve ambiguous symptoms. Moreover, *random-pattern-resistant* or *hard-to-detect* faults demand highly targeted ATPG strategies to achieve acceptable coverage [1, 6].

In parallel, machine learning (ML) has emerged as a powerful tool in electronic design automation and data analytics [7, 8]. Initial applications in IC testing demonstrate substantial benefits: Roy and his colleagues [11] reported a 20–30 % reduction in PODEM backtracks – equating to roughly 25 % faster test-pattern generation – when using neural-network–guided heuristics; Huang and his colleagues [5] achieved up to 92 % accuracy in scan-chain defect classification via support vector machines and multi-stage ANNs; Xanthopoulos and his colleagues [14] improved wafer-map die inking precision by applying RBF-SVM clustering; and Sun and his colleagues [10] nearly doubled diagnostic resolution in volume diagnosis by merging syndrome data with statistical learning. These successes, however, depend on large volumes of accurately labeled failure logs, sensitive feature engineering (e.g., SCOAP metrics, logic depth, adjacency indices), and careful integration with existing DFT and EDA toolchains – constraints that can limit practical adoption.

Given this landscape, there is mounting interest in marrying data-driven, statistical learning methods with established DFT flows to address soaring complexity and defect diversity without resorting to brute-force pattern enumeration. By leveraging predictive models and adaptable classification algorithms, test engineers can focus physical failure analysis (PFA), accelerate test-pattern generation, and reduce test escapes. The remainder of this paper provides (1) an overview of contemporary DFT strategies and their challenges; (2) a survey of ML's role in enhancing test coverage and diagnosis; and (3) a conceptual, scalable pipeline for integrating SVMs, ANNs, and feature-driven heuristics into structural test processes.

2. Overview of contemporary DFT methods and the role of machine learning

Design for Testability (DFT) refers to a suite of architectural and methodological enhancements integrated into an integrated circuit (IC) design with the primary goal of simplifying test generation, execution, and diagnosis [1, 2]. Among the principal DFT techniques are the following:

1. Scan test and Automatic Test Pattern Generation (ATPG). In a scan-based design, most sequential elements (flip-flops or latches) are arranged into one or more shift-register chains, known as *scan chains*. During test mode, each chain shifts in test data and shifts out captured responses, enabling precise control and observation of internal circuit nodes [12]. This structural test approach largely relies on ATPG, which algorithmically generates input patterns to detect modeled faults (e.g., stuck-at, bridging) at high coverage. However, as circuit complexity soars, conventional ATPG must explore exponentially growing input spaces, often leading to significant computational burdens.
2. Built-In Self-Test (BIST). BIST endows the chip with hardware to generate test stimuli and analyze

outputs internally, thereby reducing reliance on external automated test equipment (ATE) [13]. Typical BIST logic includes a linear feedback shift register (LFSR) for pseudo-random pattern generation and a multiple-input signature register (MISR) for response compaction. By localizing the test generation and analysis on-chip, BIST enables at-speed testing and can detect delay faults missed by slower external testers. BIST is extensively employed in memory subsystems (self-repair strategies) and increasingly in high-speed SoC blocks.

3. Boundary scan and board-level testing. The boundary scan standard (IEEE 1149.1) allows for a daisy-chained control of IC I/Os via a Test Access Port (TAP), providing effective board-level diagnosis without physical probing. This methodology plays a crucial role in swiftly localizing interconnect defects on complex multi-layer printed circuit boards (PCBs), especially when physical access is limited [1].

4. Memory and Analog BIST. Memory BIST targets on-chip static RAM (SRAM), dynamic RAM (DRAM), or embedded flash arrays through specialized march algorithms and redundancy repair logics (Daher, Nassar, & Youssef, 2019). Similarly, for analog or mixed-signal blocks, a suite of alternate test or sensor-based BIST has emerged – though these test insertions often demand custom analog front-ends [9].

Incorporating these DFT structures provides strong observability and controllability essential for diagnosing faults, but it also introduces design overhead (area/time). As highlighted in [9], the continual rise in transistor count and advanced packaging (e.g., multi-die systems) prompts re-evaluation of classical DFT flows, particularly in the face of novel defect mechanisms and cost constraints.

As illustrated in Thakur and his colleagues [9] (Fig. 1), one example of a bridging-defect classification flow under a structural test approach involves combining the structural netlist, simulation-based fault analysis, and dedicated DFT hooks (e.g., scan paths, controllability/observability points). This flow leads into a classification stage, which can be augmented by pattern recognition or machine learning subroutines. Such an approach underscores the central position of DFT-based instrumentation and data collection in modern test methodology.

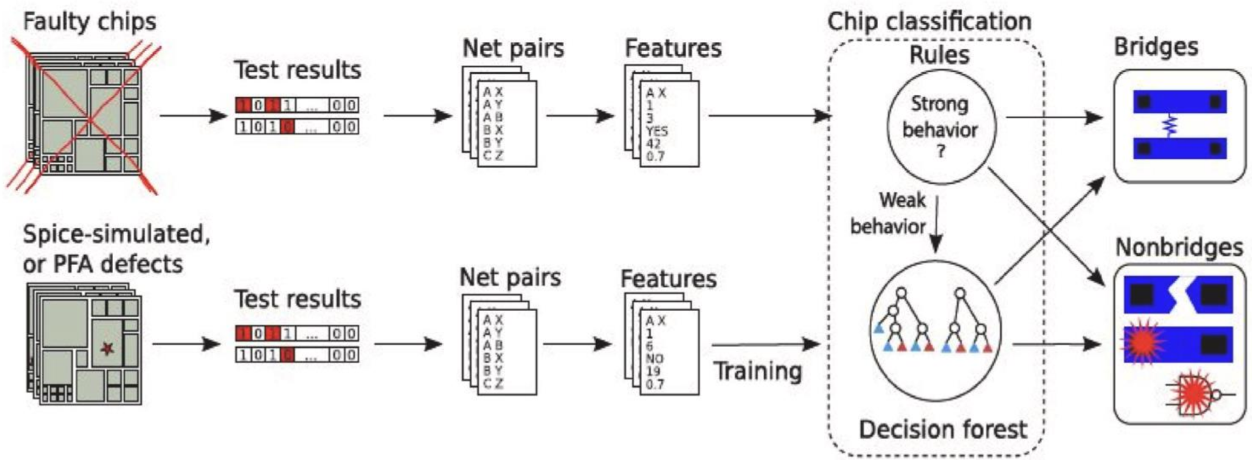

Figure 1: Flow for the classification of bridging defects [9, 15]

In that figure 1 [9], logical and physical information about potential bridging paths is integrated with output responses from scan-based tests, forming a more holistic framework for defect detection and resolution. Introducing machine learning can further refine the classification steps, especially under large-scale failure logs and diverse fault models.

In the realm of semiconductor manufacturing, a “defect” generally refers to any physical fault in the silicon or interconnect layers – bridging shorts, open vias, gate oxide breakdowns, or parametric variations – that, at the logical level, manifest as anomalies (stuck-at faults, timing hazards, bridging errors, and so forth) [6, 9]. These logical fault models guide test generation and failure analysis, but the mapping from physical defect to logical symptom is often non-trivial.

One widely used source of diagnostic information is the failure log, produced either by on-tester data collection (e.g., partial responses on a failing scan vector) or built-in test hardware. Diagnosticians may correlate repeated fail signatures to specific defect sites or defect classes [10]. Moreover, simulation-based fault injection (i.e., injecting hypothetical opens/bridges in a circuit netlist) allows engineers to cross-reference real failing patterns with precomputed signatures, thus forming the basis of volume diagnosis and yield learning [14].

Nevertheless, the major impediments in defect detection include:

- Data Volume and Complexity. Modern SoCs produce vast logs (gigabytes) of test responses, exacerbating the time needed for offline analysis.
- Diagnostic Ambiguity. Physical defects may create multiple symptom overlaps, and limited scan channels can obscure root causes, leading to suboptimal resolution [6].
- Lack of Definitive Fault Models. Variability, especially at advanced nodes (<7nm), spawns partial defects, intermittent behaviors, and soft failures that conventional stuck-at or bridging models do not fully capture [5].
- Localization vs. Repair. Identifying a single defect among billions of transistors is daunting, and post-diagnosis steps (e.g., memory row/column redundancy, device-level reconfiguration) further elevate the engineering effort [6].

Such complexity calls for solutions transcending purely algorithmic or brute-force approaches. Here, learning-based methods prove beneficial: they can extract patterns in fail logs, systematically rank suspicious regions, and adapt over time to new, emerging defect behaviors.

Recent literature extensively documents a variety of machine learning (ML) models – such as Support Vector Machines (SVMs), Random Forests, Decision Trees, Artificial Neural Networks (ANNs), and Bayesian classifiers – employed for diverse testing tasks [6, 7]. Table 1 provides a concise mapping of common test/diagnosis objectives to corresponding ML algorithms, drawing upon both [6] and complementary reviews.

Table 1: Selected ML techniques and their typical applications in electronic testing [6, 7, 9].

Testing/diagnosis objective	**Representative ML method**	**Advantages**	**Potential limitations**
Board-level functional fault diagnosis	Decision Trees, Random Forests	Fast classification; handles missing syndromes	Prone to overfitting if feature engineering is insufficient
Wafer-level clustering & die inking	K-means, SVM (with RBF kernel)	Good at identifying defect groupings on wafer maps	Sensitivity to kernel parameters; requires tuning
Scan chain defect classification (logic level)	Multi-stage ANN, Bayesian Inference	High resolution for intermittent fault localization	Complex training procedure; large dataset needed
Test-cost optimization in scan compression	Support Vector Regression (SVR)	Reduces test time by modeling cost vs. compression	Performance depends on accurate cost labeling; risk of local minima
ML-based ATPG acceleration	Neural Networks, PCA-driven search	Fewer backtracks, improved coverage/time trade-off	Must integrate with existing ATPG flow; overhead in building training set
Systematic defect detection (volume diagnosis)	Clustering + supervised classification	Identifies yield-limiting root causes quickly	May miss rare fault modes if underrepresented in training data

Several practical applications highlight how machine learning (ML) can reinforce various aspects of test and diagnosis. At the board or wafer level, ML-based classification has been employed to automate die inking; for instance, Xanthopoulos and his colleagues [14] describe a pattern recognition-driven procedure that clusters and validates failing dies spatially. Moreover, both unsupervised and supervised methods have expedited physical failure analysis, as noted by Huang and his colleagues [5].

With respect to accelerated ATPG, Roy and his colleagues [6] demonstrate that neural networks, trained on partial ATPG logs, can reduce the exponential search for test vectors by steering the backtracking algorithm toward more promising decisions. Additionally, principal component analysis (PCA) allows multiple testability metrics to be consolidated into a single guiding heuristic [11].

In the context of built-in self-test (BIST), structural signatures gathered through on-chip compression – typically via multiple input signature registers (MISR) – may be fed into machine learning modules, which can detect subtle or parametric defects overlooked by random test sequences [13]. This last approach proves especially effective for memory BIST, where the ML model rapidly isolates anomalies that arise in small-delay failure scenarios.

Overall, these ML-driven strategies suggest that static, rule-based methods are frequently inadequate when confronted with dynamically shifting defect environments and massive test data. By contrast, machine learning's ability to adapt and scale renders it as a vital component of emerging DFT solutions.

3. Proposed ML-enhanced approach for defect detection

In contemporary semiconductor manufacturing, bridging, open, and stuck-at defects frequently arise during scan-based structural testing at the wafer sort stage [9]. This phase of production typically yields detailed logs of passing and failing test patterns, partial debug outputs, and signature-based summaries, making it an advantageous point for early defect detection. Detecting defective dies at wafer sort reduces overall manufacturing costs by ensuring that inevitably failing chips do not advance further into packaging and final test [6].

The core data for such an approach usually incorporates failure logs associated with pass/fail outcomes of scan-based tests, as well as test pattern information generated by Automatic Test Pattern Generation (ATPG) or built-in self-test (BIST). Additional design context, which includes netlist structure, scan-chain parameters, and any specialized controllability/observability inserts, often complements these test logs [1]. Despite the utility of this data, its sheer scale and heterogeneous nature call for robust machine learning (ML) methods that can adapt to varying fault profiles.

A dual or hybrid machine learning strategy can be valuable in these settings. One possibility involves Support Vector Machines (SVMs) for multi-class classification tasks, especially when distinguishing bridging or open/short faults in high-dimensional feature spaces [6]. Additionally, Artificial Neural Networks (ANNs) may refine scan-test pattern generation or localize subtle and intermittent defects more effectively. Certain prior works demonstrate that neural networks can guide or prioritize suspicious nodes for test generation, thereby reducing algorithmic search overhead [11]. While other algorithms (such as Decision Trees or Random Forests) remain viable, SVMs and ANNs show particular promise in bridging-fault detection, runtime improvements for ATPG, and incremental learning of new fail modes [9].

In practice, various authors [6] have proposed conceptual pipelines integrating machine learning with scan-based DFT flows. Although specific implementation details vary, the sequence of steps follows a coherent flow from raw data collection and feature engineering to the development of machine learning models and subsequent integration with standard DFT.

One essential starting point is data consolidation, where raw failure logs from automated test equipment (ATE) at wafer sort are collected. These logs often map pass/fail status to specific test patterns, but they can also be augmented with partial BIST readings or boundary scan data when available [13]. This consolidated dataset includes design metadata, such as scan-chain configurations or controllability/observability points, which are crucial for analyzing complex fault scenarios.

An equally important step is preprocessing and feature engineering, which involves parsing the fail logs to align them with relevant structural details, such as gate-level netlists or adjacency-based bridging indices [1]. Controllability/observability metrics, such as SCOAP values and logic depth, can be fused into numeric feature vectors that capture the relative difficulty of driving or observing certain nodes. In scenarios with large volumes of data, dimensionality reduction (for instance, principal component analysis) may be used to ensure that the final ML inputs remain tractable [7].

The model development process usually centers on configuring an SVM or ANN to detect or classify bridging versus open or stuck-at defects. SVM hyperparameters can be tuned using systematic search procedures, while neural networks are often optimized via standard backpropagation with one or more hidden layers [6]. The training/validation split can follow the familiar 70%–15%–15% (training–validation–test) approach, though it may shift depending on the size and uniformity of the dataset. Model adaptation over time is also possible when new fail signatures or updated logs become available.

Finally, integration with DFT flows ensures that ML outcomes inform or improve the standard test process. For instance, an ANN might guide an ATPG engine by assigning higher priority to nodes suspected of harboring bridging faults [11]. Alternatively, a trained classifier can flag at-risk dies based on their compressed signatures, facilitating early wafer sort rejection or more detailed analysis. Table 2 summarizes key inputs, tasks, and outputs at each stage in this conceptual pipeline.

Table 2: Overview of an ML-driven methodology integrated with standard DFT steps.

Stage	**Input**	**ML Task**	**Output**
Data consolidation	Raw fail logs, netlists, optional BIST	–	Unified dataset of test logs + design features
Feature engineering	SCOAP metrics, adjacency, logic depth	–	Cleaned feature vectors (per die/test pattern)
Model development	Training set of labeled defect samples	SVM/ANN classification or regression	Optimized model(s) for wafer-level defect detection
DFT integration	Fresh fail logs + model predictions	ANN-based ATPG guidance, SVM tagging	Refined test vectors, suspect dies for deeper analysis

Previous reports have shown that bridging-fault detection often improves when adjacency-based features are included [6, 9]. SVM or ANN classifiers that incorporate net proximity indicators have demonstrated a higher recall of bridging defects. In addition, neural network models integrated into ATPG flows indicate that test-pattern generation may be shortened by reducing unproductive backtracking, as highlighted by Roy, Millican, and Agrawal [11]. Furthermore, classification-based analysis of scan-chain fail logs can improve resolution between bridging and open/short defects [5], thereby helping practitioners focus their physical failure analysis (PFA) resources.

Recent work underlines the potential for incrementally boosting bridging-fault detection by leveraging ML enhancements, relative to baseline DFT approaches without such methods [6]. Although the scale of improvement may vary across designs, the consensus is that earlier identification of subtle defect modes, especially at wafer sort, is achievable.

Moreover, combining SVMs, ANNs, or other ML models with scan-based data and structural insights can yield focused test strategies, accelerated pattern generation, and fewer test escapes. These conclusions, gleaned from multiple sources, underscore the growing preference for data-driven tactics in a domain constrained by tight cost margins, expansive fail logs, and complex defect behaviors [9]. The overall methodological outline described here provides a flexible blueprint for industrial implementations and further scholarly research into ML-based test optimizations.

4. Discussion

This work has synthesized insights from several key prior studies to outline a conceptual pipeline for integrating machine learning (ML) into Design for Testability (DFT) flows. Roy and his colleagues [11] demonstrated that artificial neural network (ANN)–guided heuristics can reduce PODEM backtracks by 20–30 %, corresponding to an approximate 25 % speed-up in test-pattern generation. Huang and his colleagues [5] reported up to 92 % accuracy in scan-chain defect classification using support vector machines (SVM) and multi-stage ANNs. Xanthopoulos and his colleagues [14] improved die-inking precision by applying RBF-SVM clustering to wafer-map failure patterns, and Sun and his colleagues [10] achieved nearly a two-fold increase in diagnostic resolution for volume diagnosis by merging syndrome data with statistical learning.

Our proposed framework unites these advances by consolidating raw failure logs and DFT metadata, extracting predictive features (e.g., SCOAP values, logic depth, adjacency metrics), training SVM/ANN classifiers, and feeding their inferences back into scan-based ATPG or yield-management systems. In doing so, it offers a unified, scalable approach that can adapt to diverse fault models without exhaustive pattern enumeration – addressing the computational burdens of conventional ATPG [3].

However, several limitations and constraints warrant careful consideration:

1. Conceptual Scope. The present work remains a high-level framework without new empirical validation on industrial-scale datasets. Its efficacy must be confirmed through pilot studies using real wafer-sort logs and scan-chain data.
2. Data Availability and Label Scarcity. Robust ML training requires large volumes of accurately labeled failure logs. In many production environments, such labels may be sparse or inconsistently recorded, necessitating semi-supervised or data-augmentation techniques.
3. Feature-Engineering Sensitivity. The predictive power of SVM and ANN models depends critically on

quality features. While SCOAP, logic depth, and adjacency metrics are proven starting points [9], emerging defect modes at sub-7 nm nodes may require novel topology- or layout-based descriptors.

4. Integration Overhead. Embedding ML modules into existing EDA toolchains and DFT architectures introduces both area and runtime costs. Real-time or in-field applications (e.g., BIST-based anomaly detection) must balance model complexity against on-chip resource constraints.
5. Evolving Fault Signatures. Semiconductor processes continually evolve, producing new defect mechanisms. ML models must be periodically retrained or extended to accommodate emerging fault classes, which implies an ongoing maintenance effort.

By candidly acknowledging these constraints, we aim to provide a realistic assessment of the path forward. Our survey of previous results demonstrates clear potential for ML-augmented DFT, yet underscores the necessity of targeted validation, robust data pipelines, and tight integration with manufacturing test infrastructures.

5. Conclusion & Future Work

The relentless march of technology scaling and the resulting proliferation of intricate fault modes have outpaced the capabilities of purely heuristic-based DFT methods. While scan chains, boundary scan, and BIST remain foundational, their efficacy can be substantially enhanced through data-driven ML techniques. Prior studies – spanning ANN-guided ATPG, high-accuracy defect classification, wafer-map clustering, and volume diagnosis improvements – validate the premise that learning-based models can reduce test time by 20–30 %, achieve over 90 % classification accuracy, and double diagnostic resolution.

Our conceptual pipeline integrates these insights into a cohesive workflow: consolidating raw fail logs and DFT metadata; engineering features reflective of controllability, observability, and topology; training SVM and ANN classifiers; and leveraging their outputs to guide ATPG and yield-management decisions. This framework promises more targeted test vectors, accelerated pattern generation, and fewer test escapes.

Nevertheless, realizing this promise in production requires overcoming tangible challenges: securing sufficiently large and accurately labeled log datasets; engineering features attuned to novel nanoscale defect mechanisms; embedding ML modules within area- and power-constrained environments; and establishing processes for ongoing model retraining as fault signatures evolve.

Future research should therefore pursue:

- Empirical Validation. Implement the pipeline on industrial test benches to quantify gains in test time, coverage, and diagnostic resolution.
- Advanced Learning Architectures. Explore graph neural networks to natively capture circuit topology, and deep learning for end-to-end test optimization.
- Cross-Layer Co-Optimization. Integrate physical-layout awareness into ML features, enabling simultaneous optimization of design and test.
- In-Field Adaptation. Develop lightweight, on-chip inference engines for continuous BIST-driven monitoring and anomaly detection.

By addressing these directions, the ML-guided DFT paradigm can mature into a robust, scalable solution – ensuring that test and diagnosis keep pace with the growing complexity of next-generation semiconductor devices.

References

[1]. Bushnell, M., & Agrawal, V. (2004). *Essentials of electronic testing for digital, memory and mixed-signal VLSI circuits* (Vol. 17). Springer Science & Business Media.

[2]. Wang, L. T., Wu, C. W., & Wen, X. (2006). *VLSI test principles and architectures: design for testability*. Elsevier.

[3]. Cheng, K. T., & Agrawal, V. D. (1989). *Unified methods for VLSI simulation and test generation*. Kluwer Academic Publishers.

[4]. Moore, G. E. (1998). Cramming more components onto integrated circuits. *Proceedings of the IEEE*, *86*(1), 82-85.

[5]. Huang, Y., Benware, B., Klingenberg, R., Tang, H., Dsouza, J., & Cheng, W. T. (2017, November). Scan chain diagnosis based on unsupervised machine learning. In *2017 IEEE 26th Asian Test Symposium (ATS)* (pp. 225-230). IEEE.

[6]. Roy, S., Millican, S. K., & Agrawal, V. D. (2024). A Survey and Recent Advances: Machine Intelligence in Electronic Testing. *Journal of Electronic Testing*, *40*(2), 139-158.

[7]. Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction* (Vol. 2, pp. 1-758). New York: springer.

[8]. Stratigopoulos, H. G. (2018, May). Machine learning applications in IC testing. In *2018 IEEE 23rd European Test Symposium (ETS)*(pp. 1-10). IEEE.

[9]. Thakur, G., Jain, S., & Sohal, H. (2022). Current issues and emerging techniques for VLSI testing-A review☆. *Measurement: Sensors*, *24*, 100497.

[10]. Sun, Z., Jiang, L., Xu, Q., Zhang, Z., Wang, Z., & Gu, X. (2015, January). On test syndrome merging for reasoning-based board-level functional fault diagnosis. In *The 20th Asia and South Pacific Design Automation Conference* (pp. 737-742). IEEE.

[11]. Roy, S., Millican, S. K., & Agrawal, V. D. (2021, May). Unsupervised learning in test generation for digital integrated circuits. In *2021 IEEE European Test Symposium (ETS)* (pp. 1-4). IEEE.

[12]. Abramovici, M., Breuer, M. A., & Friedman, A. D. (1990). *Digital systems testing and testable design* (Vol. 2, pp. 203-208). New York: Computer science press.

[13]. Stroud, C. E. (2005). *A designer's guide to built-in self-test* (Vol. 19). Springer Science & Business Media.

[14]. Xanthopoulos, C., Sarson, P., Reiter, H., & Makris, Y. (2017, October). Automated die inking: A pattern recognition-based approach. In *2017 IEEE International Test Conference (ITC)* (pp. 1-6). IEEE.

[15]. Nelson, J. E., Tam, W. C., & Blanton, R. D. (2010, November). Automatic classification of bridge defects. In *2010 IEEE International Test Conference* (pp. 1-10). IEEE.