Pre-processing Functional And Physical Defect Equivalences To Accelerate Cell-Aware Model Generation

Pre-processing Functional And Physical Defect Equivalences To Accelerate Cell-Aware Model Generation

***Abstract*—Detecting intra-cell defects accurately is essential to ensure rapid yield ramp-up and minimize test escapes. Functional fault models like stuck-at and transition faults do not fully capture the complexity of these transistor-level defects. Cell-aware Test (CAT) was developed to address this, but its characterization phase has a significant runtime cost. This paper presents a novel pre-characterization methodology that reduces CAT characterization time by identifying intra-cell transistor defects equivalent to functional faults. By applying structural equivalence and fault propagation rules, the proposed approach reduces the Cell-Aware defect count to be simulated by removing defects from the list which are proven to be equivalent to functional faults, while keeping track of the equivalent defects for diagnosis purposes. Validation on two different 28nm and a 130nm industrial technology libraries demonstrated an overall reduction of 25% in defect count requiring analog characterization, resulting in an overall runtime reduction of about 7.3%. This method reduces the CAT workload, accelerating the time-consuming fault characterization step, without compromising defect detection or diagnosis accuracy.**

***Index Terms*—intra-cell defects, functional faults, stuck-at faults, transition faults, cell-aware test model, optimization**

I. Introduction

The increasing complexity of integrated circuits [1] has driven the need for advanced testing methodologies to detect a wider range of manufacturing defects. Cell-Aware Test (CAT) [2]–[5] effectively addresses defects within standard cells, providing an improved defect coverage compared to functional fault models such as stuck-at and transition faults [6], [7]. By representing physical defects like pure transistor opens, shorts, and bridging defects, CAT improves the effectiveness of test patterns for intra-cell defect detection and diagnosis. A critical step in CAT is characterization, where all input stimuli combinations are applied to the cell's netlist and each intra-cell defect is injected to evaluate its impact, generating a Cell-Aware Model (CAM) for digital test tools such as ATPG or fault simulators. However, this step requires intensive computation due to the large number of analog simulations needed [8], making runtime reduction a key challenge for large technology libraries. Prior methodologies have been introduced to speed up CAM generation. In [9], a machine learning flow predicts defect detectability, achieving over 99% runtime reduction for many cells while maintaining accuracy. Authors in [10] use structural analysis to identify logically undetectable defect-pattern pairs, excluding them before simulation and cutting simulation workload by nearly half. In [11], [12] authors proposed TrUnDeL, a graph-based method that excludes undetectable defect-pattern pairs, reducing CAM generation time by 30% for sequential cells and 60% for combinational cells. Authors in [13] also leveraged voltage information from defect-free analog simulations to improve CAM generation efficiency. In [14] authors propose to use the equivalency between Cell-Aware (CA) defects and functional fault models to speed up CA fault simulation.

While prior work has focused on excluding undetectable defects, the equivalence between detectable CA defects and functional fault models has not been studied. We show that exploiting this equivalence can further reduce the computational cost of CAM generation.

This paper introduces a pre-characterization optimization methodology, called *EquivalentID*, that systematically identifies transistor defects equivalent to functional faults and excludes them from the SPICE simulation queue. This method is implemented using lightweight Python scripts which is applicable in a matter of seconds. The excluded defects are logged in a diagnosis file to ensure their information remain available for fault localization and analysis. We demonstrated the effectiveness of this approach through application on two different 28nm and a 130nm industrial libraries, achieving a 7.3% workload reduction for CAM generation by identifying 25% of transistor defects equivalent to functional faults. This methodology complements approaches that exclude undetectable defect–pattern pairs by also removing equivalent defects from the detectable set, enabling the equivalent intra-cell transistor faults to be processed with faster fault simulation instead of slower analog simulation.

The rest of the paper is organized as follows: Section II provides an overview for CAT and transistor defect modeling. In Section III we introduce our proposed methodology to identify equivalencies. Then in Section IV we present experimental results and the validation process to verify the effectiveness of the proposed approach. Finally in Section V we draw conclusions and discuss possible future works.

II. Background

A. Cell-Aware Test

CAT is an advanced testing methodology that models physical defects within standard cells (i.e., intra-cell defects), enabling more accurate detection of manufacturing defects than functional fault models. Unlike classical functional faults that assume idealized digital behavior, CAT considers intra-cell physical defects such as pure transistor opens, shorts between gate, source, and drain terminals, and bridging faults (i.e.,

1MΩ for opens and 1Ω for shorts). CA defects can fall into two categories according to their specifications. *Static CA defects* which are defects whose their detection require a single pattern (1-timeframe pattern) and *Dynamic CA defects* which are detectable using a sequence of two patterns (2-timeframe pattern).

By targeting intra-cell defects, CAT improves defect coverage and diagnostic resolution, especially for suitable defects that may not manifest as functional faults at the gate level. This fine-grained modeling, however, results in a significantly larger defect set per cell, increasing the complexity and runtime of subsequent characterization.

B. Transistor Defect Modeling

In CAT, each transistor within a cell is modeled to have multiple possible defect types, typically considering nine realistic fault mechanisms caused by gate-to-source short, gate-to-drain short, source open, drain open, and others as represented in Fig. 1. These defects are characterized by their physical location on the transistor and their electrical impact on the cell's behavior. Characterizing these defects involves simulating their fault effects across all relevant input combinations to generate CAMs used in ATPG, fault simulation, and diagnosis. This process is computationally intensive (i.e., CPU time and license count), as each defect requires detailed SPICE simulation to accurately capture analog and timing effects.

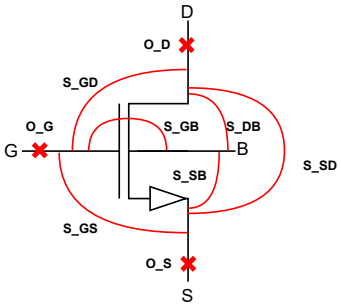


Fig. 1: 9 possible transistor defect models with (*O_*) representing open defects and (*S_*) referring to short defects

C. Compartments in Complex Cells

Modern standard cells often consist of multiple internal functional blocks, or *compartments* [10], [11], which are logical groupings of transistors implementing sub-functions within the cell. For example, Fig. 2 illustrates compartments of a complex cell. These compartments communicate through intermediate nodes not exposed as primary inputs or outputs, causing fault effects originating in one compartment to propagate through others before reaching the cell's primary outputs. Understanding this hierarchical structure is essential for accurate defect propagation comprehension, as fault effects can be masked or transformed as they propagate throughout upcoming compartments. Consequently, equivalency analysis must consider how defects impact downstream compartments to determine their observability at cell's primary outputs.

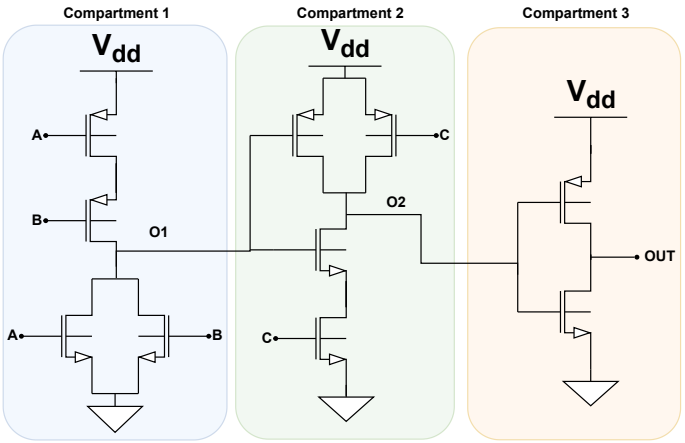


Fig. 2: Compartments of a complex cell

III. Methodology

The proposed methodology identifies transistor defects equivalent to functional faults (i.e., stuck-at and transition faults) and excludes them from costly SPICE simulations, while preserving diagnostic information via a dedicated diagnosis log. An example of this concept is shown in Fig. 3.

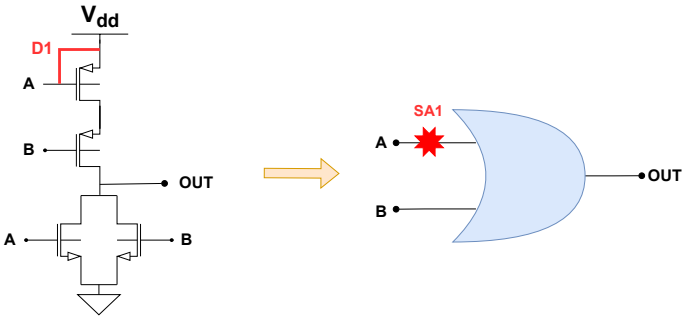


Fig. 3: Transistor defect D1 has the same fault effect as the SA1 fault on the primary input A

The methodology is structured around three core equivalency rules applied to transistor defects extracted from the transistor-level netlist of each cell in the technology library. Fig. 4 gives an overview of the methodology application. The program called *EquivalentID.py (Equivalency Identifier)* receives the *.spi netlist* containing transistor-level structure information of the cells and a *transistor defect list* which contains information on all the intra-cell transistor defects. Then, it applies each equivalency rule independently and identifies the equivalent defects. Finally, it produces a *reduced defect list* for the defects that are proven not to have any equivalence with functional faults and a *functional equivalent defect list* log file containing information of transistors defects equivalent to functional faults.

The structural and propagation rules applied to identify the equivalencies are detailed in the following:

A. Rule 1: Structural Equivalence

Structural equivalence identifies transistor defects that can be directly mapped to stuck-at faults at the primary inputs and

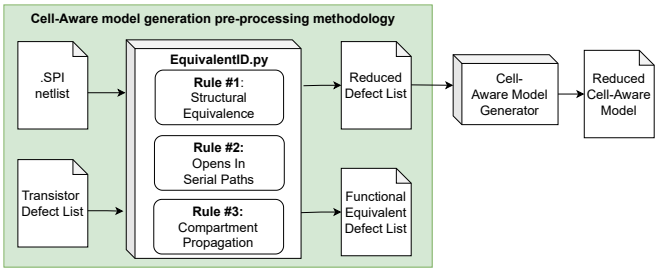


Fig. 4: Application of the methodology

outputs (I/Os) of the cell by analyzing the transistor's physical connectivity and the association of its terminals to cell pins and power rails.

The core idea is to detect short defects occurring between two terminals of a transistor where one terminal is connected to a primary I/Os pin of the cell, and the other terminal is connected to the power rail (either V_{DD} or GND). This configuration effectively forces the logic value at the I/O pin of the gate to a constant logic level, mimicking the behavior of a stuck-at fault. There can be two cases that can occur under such condition:

- If the transistor's I/O pin is shorted to GND, the defect forces the node to a logical '0' regardless of the intended signal. This fault effect is equivalent to a stuck-at-0 ($SA0$) fault on the corresponding cell pin.
- Conversely, if the transistor's I/O pin is shorted to power supply (V_{DD}), the defect forces the node to a logical '1', producing a fault effect equivalent to a stuck-at-1 ($SA1$) fault on the cell pin.

This equivalency holds because the short defect imposes a fixed logic level at the pin, overriding normal circuit operation and replicating the stuck-at fault behavior at the logical level.

Formally, if we consider P_{IO} as the transistor pin connected to the cell's primary I/O, P_{Power} as the transistor pin connected to a power rail, either V_{DD} or GND, and D_{short} as the defect representing a short between P_{IO} and P_{Power}, the fault equivalency can be expressed as: $D_{\text{short}}(P_{\text{IO}}, GND) \equiv SA0$ at P_{IO} and $D_{\text{short}}(P_{\text{IO}}, V_{DD}) \equiv SA1$ at P_{IO} where "$\equiv$" denotes equivalence in fault effect on the logical behavior of the cell.

Fig. 5 illustrates these conditions using the transistor-level schematic of an inverter. In this example, defect $D1$ corresponds to a short between the gate and drain terminals of transistor $MP1$, resulting in a $SA1$ fault effect on the primary input A. Similarly, defect $D2$ represents a short between the drain and source terminals of transistor $MN1$, producing a $SA0$ fault effect on the primary output OUT.

B. Rule 2: Opens In Serial Paths

Rule 2 focuses on identifying equivalences between pure open transistor defects and transition faults (i.e. slow-to-rise (STR) and slow-to-fall (STF)). These defects occur within serial transistor paths inside a standard cell. The serial paths are defined specifically as the chains of transistors connecting a primary output pin of the cell to one of the power rails. The fundamental insight behind this rule is that an open defect in such a serial path effectively disconnects the output node

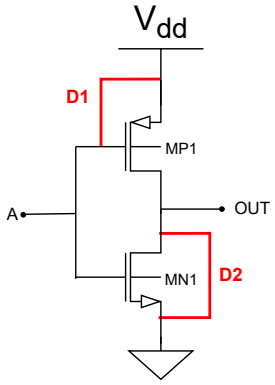


Fig. 5: Transistor-level schematic of an Inverter

from V_{DD} or GND affecting its logic drive capability, and constraining it to fewer valid digital values [15], [16].

- If the serial path to V_{DD} is inaccessible because of an open defect, the output node can only be pulled down or at floating state, as the concerned part of the circuit loses its pull-up capability.
- Conversely, if the serial path to GND is broken, the output node is limited to pull-up or floating state.

Detecting such defects requires applying 2-timeframe patterns [17]. For example, consider a NOR gate with an injected open defect (e.g., defect $D3$), as illustrated in Fig. 6. This defect can only be detected using a 2-timeframe pattern where the first cycle sets the cell output to '0', and the second cycle attempts to set the output to '1'. Due to the presence of defect $D3$, the NOR gate loses its ability to drive the output to '1', causing the output to retain the previous cycle's value ('0') and making the fault effect detectable.

In terms of equivalencies, it means that these defects are equivalent to transition faults where the first pattern cycle is used to set the output at a given state and the second pattern cycle is used to detect the fault. Specifically, in the example defect $D3$ can be considered as equivalent to a STR transition fault in which the output never goes to '1' as we are considering pure open defects.

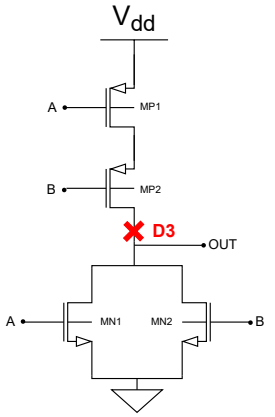


Fig. 6: Transistor-level schematic of a 2-input NOR logic gate

In formal terms, if we consider OUT as the *primary output pin* of the cell, V_{DD} and GND as the power supply rails, a *serial transistor path* S as an ordered sequence of transistors connecting the primary output OUT to either V_{DD} or GND, and $D_{\text{open}}(S_i)$ as an open defect occurring on the i^{th} transistor in the serial path S, we can conclude the fault effect equivalencies are expressed as:

$$D_{\text{open}}(S_i) \equiv \begin{cases} \text{STR at } OUT, & \text{if } S \text{ connects } OUT \text{ to } V_{DD} \\ \text{STF at } OUT, & \text{if } S \text{ connects } OUT \text{ to } GND \end{cases}$$

In order to apply rule 2, it is essential to first recognize the *parallel branch structures* present in the pull-up and pull-down networks. Parallel branches introduce alternative paths between the primary output and a power supply rail, which can mask the effect of an open defect in one branch.

To manage such cases, we classify parallel branches into three categories:

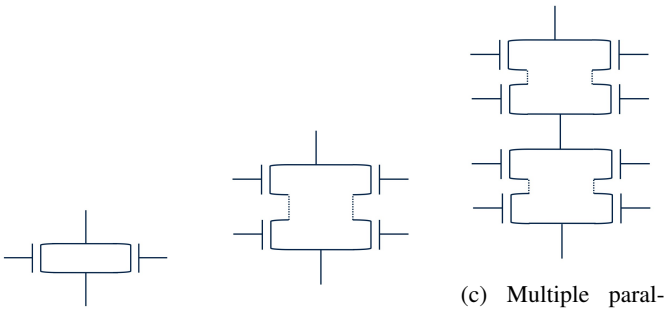
- *Direct Parallel Branch:* Two individual transistors are directly connected in parallel between the same pair of nodes as presented in Fig. 7a. An open defect in one transistor may have no observable effect if the parallel transistor remains functional, as current can still flow through the alternative path.
- *Parallel Branch of Series Transistors:* Two or more series-connected transistor chains placed in parallel between the same nodes illustrated in Fig. 7b. An open defect in one transistor within one branch is masked if the other path in the branch maintains connectivity.
- *Multiple Parallel Branches in Series with Other Parallel Branches*: As shown in Fig. 7c, this is the generalized version of the previous case which represents complex network structures where a set of parallel branches is itself connected in series with another set of parallel branches. These configurations provide multiple redundancy points, further reducing the likelihood of an open defect fully disconnecting the output from the power supply rail.

These parallel branch structures are considered when identifying the serial paths and are not considered as part of the serial paths. By removing them from the analysis, we ensure that only *true single conductive paths*—where an open defect directly serves the connection between the primary output and the power supply—are analyzed.

C. Rule 3: Compartment-to-Compartment Fault Propagation

Rule 3 establishes the equivalence between transistor defects (i.e., open, short) and functional fault models (i.e., stuck-at, transition). While Rules 1 and 2 address transistor defects that directly impact the primary I/O of a cell and map to functional faults at the pin level or along serial transistor paths, many complex standard cells are composed of multiple internal functional blocks, which we refer to as *compartments* see Section II-C. Each compartment implements part of the cell's logic function and communicates with others through intermediate nodes that are not exposed as primary I/Os.

Rule 3 extends the defect equivalence analysis to such multi-compartment cells by applying the principles of rules 1 and 2 to intermediate compartments and then determining whether the resulting fault effects can propagate to a primary output.



(a) Direct parallel branch

(b) Parallel branch with series of transistors

(c) Multiple parallel branch in series with another parallel branch

Fig. 7: Cases of parallel branches

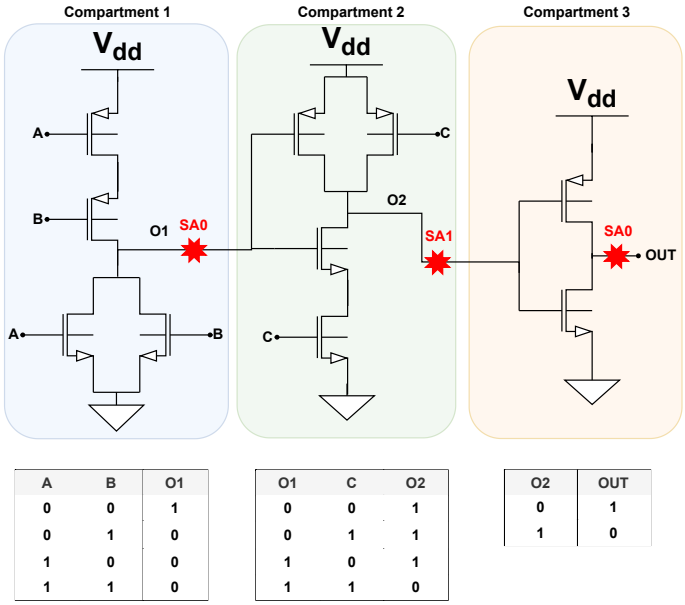
In a single-compartment cell (e.g., a NAND gate), Rules 1 and 2 are sufficient because all transistor networks connect directly to primary pins. However, in a multi-compartment cell, only some compartments directly interface with the cell's primary I/Os.

The methodology relies on the truth table of each compartment to evaluate whether a fault effect originating inside one compartment will propagate through the downstream compartments to the primary output. This step models the internal logic flow, assessing whether the defect's logical faulty effect remains observable after functional transformations such as inversion, masking, or logic combination.

If the analysis determines that the defect's fault effect has been propagated to the primary output—regardless of the external input combination—it is deemed equivalent to a functional fault at that primary output. In case a defect equivalence with stuck-at fault is generated by rule 1 and propagated to the output, it means that this defect is equivalent to a stuck-at fault on the output. On the other hand, if the defect is identified as equivalent to a transition fault by rule 2, it results in a transition fault equivalency on the primary output if propagated.

Fig. 8 illustrates a fault effect propagation example through the compartments of a complex cell with primary I/Os $[A, B, C, OUT]$. The methodology also manages more than one fan-out between compartments. It is sufficient to propagate the fault effect along each branch of the fan-out until it reaches a primary output. For the sake of simplicity, we present an example with single fan-out. A short transistor defect causing a $SA0$ fault effect on the intermediate node *O1* would not be detected by the previous rules when considering isolated compartments. However, analyzing *compartment 1* individually reveals multiple transistor defects that produce this $SA0$ effect on *O1* identified by rule 1. By examining the truth table of *compartment 2*, we observe that a logic '0' on any of its inputs results in a $SA1$ effect on its output. Furthermore, with a constant logic '1' applied to the input of *compartment 3*, the $SA0$ effect on *O1* propagates as a stuck-at-0 fault on the primary output OUT of the cell. Consequently, any transistor

defect identified by rule 1 producing a $SA0$ effect on the intermediate node $O1$ is considered equivalent to a $SA0$ fault at the primary output OUT of the cell.



A	B	O1
0	0	1
0	1	0
1	0	0
1	1	0

O1	C	O2
0	0	1
0	1	1
1	0	1
1	1	0

O2	OUT
0	1
1	0

Fig. 8: Example of fault effect propagation in a complex cell

In essence, Rule 3 generalizes Rules 1 and 2 to cover fault effects in internal nodes of complex cells by verifying their ability to propagate and manifest as functional faults at primary outputs.

D. Functional Equivalent Defect List

A critical aspect of our pre-characterization optimization methodology is the preservation of diagnostic information for transistor defects excluded from SPICE simulation. Although these defects are identified as equivalent to functional faults and removed from the characterization workload, their presence and characteristics remain essential for accurate fault localization and diagnosis during testing.

To address this, a comprehensive diagnosis log file is generated that records all transistor defects classified as equivalent to functional faults. This log serves as a repository of defect metadata, ensuring that no diagnostic information is lost due to runtime optimization. Each logged defect is traceable to its original transistor-level location, defect type, and their identified equivalent functional fault is also reported which enables detailed analysis and correlation with test outcomes.

IV. Experimental Results

In order to assess the effectiveness of the proposed methodology, we used two different FD-SOI 28nm and a 130nm industrial standard cell libraries (i.e., P28, C28, and C9A). We used python to develop the tool to apply the equivalency rules mentioned in Section III. We also used a commercial tool to characterize the libraries and produce the CAMs. To extract the compartment information in order to apply rule 3, we used an in-house custom tool. In the following, we will exhibit results obtained on the reduction achieved on the number of transistor defects in the SPICE simulation queue as well as the acceleration in runtime after applying the methodology.

A. Equivalent Defects Analysis

Table I represents the number of transistor defects present in the SPICE simulation queue before and after applying the methodology. Originally, in order to characterize the combinational cells in the libraries under analysis, a high number of transistor defects relative to the library size are to be simulated and characterized. Using the proposed equivalency rules, around 25% of transistor defects are identified as equivalent to functional faults and can be excluded from the list of defects to be characterized. Results for the C9A library lead to a slightly lower reduction which is due to the presence of large cells with a few number of primary I/Os that impacts the effectiveness of rule 1.

In order to have a better understanding, we analyzed the contribution of each rule both in the total number of identified equivalent defects and across different transistor size in the P28 library. The pie chart in Fig. 9 exhibits the contribution of each rule in the 25% defect count reduction for the P28 library. It is shown that out of the 56,500 equivalent defects, 86% of them were identified thanks to rule 1 followed by rule 3 and rule 2. This breakdown provides insight into the relative impact of each rule on runtime optimization, emphasizing that while structural equivalence alone offers significant savings, the combined application of all three rules maximizes defect reduction.

TABLE I: Number of transistor defects to characterize before and after pre-processing

Library	Initial Defects	With pre-processing	Reduction Ratio
P28	221,004	164,505	25.30%
C9A	99.306	77,790	21.65%
C28	51,750	38,053	26.47%

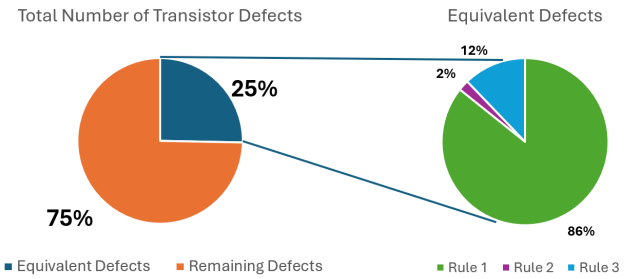


Fig. 9: Total equivalent defect detection percentage per rule in P28

To investigate the impact of cell size and complexity on the number of equivalent defects identified by the methodology, we categorized the cells in the P28 library based on their transistor count. The graph in Fig. 10 illustrates the contribution of each equivalency rule within these groups, along with the

overall percentage of equivalency detection relative to cell size. Rule 1 consistently identifies the largest share of equivalent defects even in complex cells where we expect relatively fewer defects connected to I/O pins. In contrast, Rule 2 significantly contributes in smaller cells but its impact diminishes rapidly as cell complexity grows, primarily due to the presence of parallel transistor branches that mask open defects. Rule 3 offers a steady, moderate contribution mainly in medium to large cells, reflecting its role in capturing equivalencies stemming from internal compartmental structure and fault propagation. Overall, the total percentage of equivalent defects is highest in smaller cells with fewer than 10 transistors, while it stabilizes between 20% and 30% for larger, more complex cells. This analysis demonstrates that although equivalency detection decreases initially with increasing cell size, the methodology remains robust and effective regardless of cell complexity. Libraries C28 and C9A also exhibited similar behavior.

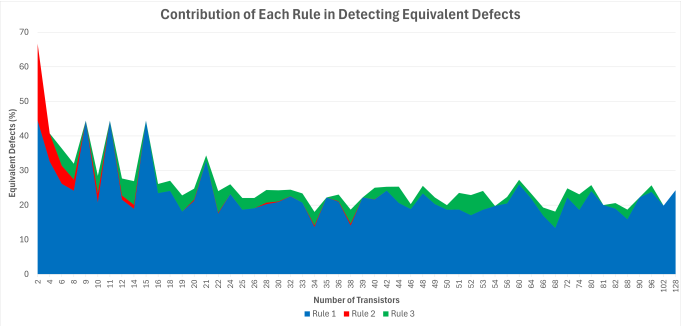


Fig. 10: Rule contribution to number of equivalent defects per cell size in P28

B. Runtime Acceleration

TABLE II: Characterization Runtime in hours

Library	Baseline	Optimized	EquivalentID	Reduction
P28	2,208 h (92 d)	2,042 h (85 d)	125 s	7.52%
C9A	975 h (41 d)	910 h (38 d)	86 s	6.64%
C28	508 h (21 d)	469 h (20 d)	54 s	7.69%

The runtime comparison represented in Table II between the baseline and optimized characterization flows demonstrates that by applying the equivalentID method to the transistor defect set, which only takes a matter of seconds (i.e., 125 seconds for P28, 86 seconds for C9A and 54 seconds for C28), we achieve an overall runtime reduction of approximately 7.3% relative to the total characterization time for the full transistor defect set. This percentage translates to a substantial absolute savings of about 39 hours to 7 days on a baseline runtime exceeding 20 days to three months (92 days) relative to the size of the library. This reduction mostly stems from optimizations in static defect set where the total characterization runtime consists of both static and dynamic defect simulations. The results also prove that the effectiveness of the methodology remains when the library size grows as the reduction ratio holds regardless of the library size. Given the computationally intensive nature of SPICE simulations and the scale of modern technology libraries, this reduction represents a significant acceleration in the characterization process. Importantly, this runtime improvement is achieved without compromising the accuracy or diagnostic completeness of the CAT models. The other significant fact to point out is that the proposed methodology is implemented with an efficient Python tool that executes in a matter of seconds. This rapid pre-processing step results in nearly a full week of saved SPICE simulation time during the characterization phase.

C. Validation

To rigorously validate the correctness of the identified equivalencies between transistor defects and functional faults, we employed the methodology described in [14] called *FACT*. This approach provides a systematic framework to extract equivalency correlations directly from the CAMs generated without any post-processing optimizations.

Using FACT, we obtained a reference mapping of transistor defects to their corresponding functional faults based on the original, fully characterized CAMs. We then performed a detailed comparison between these reference equivalencies and those identified by the proposed methodology. This comparison was conducted at the granularity of individual defects, ensuring that both the exact defect instances and their correlated functional fault types matched. The validation results demonstrated a complete agreement between the equivalencies identified by the two approaches, with no mismatches observed.

V. Conclusions

This paper presented a novel pre-characterization optimization methodology, called *EquivalentID*, that identifies transistor defects equivalent to functional faults and excludes them from costly SPICE simulations during CAT library characterization with a negligible application time. By applying a set of structural and fault propagation equivalency rules, the methodology effectively reduces the number of defects requiring detailed characterization while preserving full diagnostic coverage through a comprehensive diagnosis log. Experimental results on two different 28nm and a 130nm industrial technology library demonstrate an average of 25% reduction in transistor defects to be characterized, translating into an overall runtime reduction of approximately 7.3%. Given the computational intensity of transistor-level simulations, this runtime improvement is significant and enables faster CAT library development while preserving model accuracy and diagnostic fidelity. The methodology's implementation as a lightweight Python tool ensures rapid execution, making it a practical and scalable solution for modern technology libraries. Future work will focus on extending the approach to more equivalency rules to cover more dynamic defects and sequential cells to further speed up characterization since dynamic defect simulation owns the largest part of the total characterization runtime.

REFERENCES

[1] M. G. Bardon, Y. Sherazi, P. Schuddinck, D. Jang, D. Yakimets, P. Debacker, R. Baert, H. Mertens, M. Badaroglu, A. Mocuta, N. Horiguchi, D. Mocuta, P. Raghavan, J. Ryckaert, A. Spessot, D. Verkest, and A. Steegen, "Extreme scaling enabled by 5 tracks cells: Holistic design-device co-optimization for finfets and lateral nanowires," in *2016 IEEE International Electron Devices Meeting (IEDM)*, 2016.

[2] F. Hapke, R. Krenz-Baath, A. Glowatz, J. Schloeffel, H. Hashempour, S. Eichenberger, C. Hora, and D. Adolfsson, "Defect-oriented cell-aware atpg and fault simulation for industrial cell libraries and designs," in *International Test Conference*, 2009.

[3] F. Hapke, W. Redemund, A. Glowatz, J. Rajski, M. Reese, M. Hustava, M. Keim, J. Schloeffel, and A. Fast, "Cell-aware test," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 9, 2014.

[4] F. Hapke, W. Redemund, J. Schloeffel, R. Krenz-Baath, A. Glowatz, M. Wittke, H. Hashempour, and S. Eichenberger, "Defect-oriented cell-internal testing," in *IEEE International Test Conference*, 2010.

[5] F. Hapke, M. Reese, J. Rivers, A. Over, V. Ravikumar, W. Redemund, A. Glowatz, J. Schloeffel, and J. Rajski, "Cell-aware production test results from a 32-nm notebook processor," in *IEEE International Test Conference*, 2012.

[6] K. Y. Cho, S. Mitra, and E. McCluskey, "Gate exhaustive testing," in *IEEE Int'l Conf. on Test*, 2005.

[7] S. Eichenberger, J. Geuzebroek, C. Hora, B. Kruseman, and A. Majhi, "Towards a world without test escapes: The use of volume diagnosis to improve test quality," in *2008 IEEE International Test Conference*, 2008.

[8] Z. Gao, S. Malagi, M.-C. Hu, J. Swenton, R. Baert, J. Huisken, B. Chehab, K. Goossen, and E. J. Marinissen, "Application of cell-aware test on an advanced 3nm cmos technology library," in *2019 IEEE International Test Conference (ITC)*, 2019.

[9] P. D'Hondt, A. Ladhar, P. Girard, and A. Virazel, "A comprehensive learning-based flow for cell-aware model generation," in *IEEE International Test Conference (ITC)*, 2022.

[10] F. Lorenzelli, Z. Gao, J. Swenton, S. Malagi, and E. J. Marinissen, "Speeding up cell-aware library characterization by preceding simulation with structural analysis," in *IEEE European Test Symposium (ETS)*, 2021.

[11] G. Mongelli, E. Faehn, D. Robins, P. Girard, and A. Virazel, "A fast and efficient graph-based methodology for cell-aware model generation," in *2024 IEEE International Test Conference (ITC)*, 2024.

[12] G. Mongelli, E. Faehn, D. Robins, P. Girard, and A. Virazel, "Accelerating cell-aware model generation for sequential cells using graph theory," in *2025 Design, Automation Test in Europe Conference (DATE)*, 2025.

[13] C. Chen, W. Zhang, N. Wang, and Y. Huang, "Improving efficiency of cell-aware fault modeling by utilizing defect-free analog simulation," in *2023 International Symposium of Electronics Design Automation (ISEDA)*, 2023.

[14] R. Khoshzaban, I. Guglielminetti, M. Grosso, M. S. Reorda, and R. Cantoro, "Exploiting the correlation with traditional fault models to speed-up cell-aware fault simulation," in *2025 IEEE International Test Conference (ITC)*, 2025.

[15] D. Ronga, G. Mongelli, E. Faehn, P. Girard, and A. Virazel, "Analog to digital memory modeling for test," *Microprocessors and Microsystems*, vol. 118, 2025.

[16] D. Ronga, X. Xhafa, E. Faehn, P. Girard, T. Vayssade, and A. Virazel, "A digital sram modeling for cell-aware testing and test algorithms evaluation," in *2025 IEEE 31st International Symposium on On-Line Testing and Robust System Design (IOLTS)*, 2025.

[17] J. C.-M. Li and E. McCluskey, "Diagnosis of resistive-open and stuck-open defects in digital cmos ics," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 11, 2005.