A Comprehensive Scan Test Cost Model to Optimize the Production of Very Large SoCs

Giusy Iaria, *Member, IEEE*, Paolo Bernardi, *Senior Member, IEEE*, Claudia Bertani, Lorenzo Cardone, *Graduate Student Member, IEEE*, Giuseppe Garozzo, and Vincenzo Tancorre

***Abstract*—This paper explores the trade-offs of reducing scan test patterns during Wafer Sort, accepting additional packaging costs, and screening more chips during Package Tests. Previous works proposed ways of selecting or reordering patterns to bring the most efficient to the left. Unlike such studies, this work quantifies the benefit of removing patterns directly from the tail of any pattern set. The paper elaborates on novel formulas to propose a comprehensive cost model that combines yield, Wafer Sort, packaging, and Package Test costs. The model evolves from known concepts by assuming that mass production defectivity is non-uniformly distributed over the die population and accounts for sacrificial lots to extract guiding information. It is shown that reducing patterns at Wafer Sort is beneficial under certain conditions of yield, fault coverage, and considering equipment and production costs. The model accurately estimates the number of patterns to remove for maximum gain in these cases. As a further by-product, the paper shows that a significant cost advantage can be achieved if pattern generation is guided based on the basics of the non-uniform failure distribution. This approach is validated with an academic benchmark and by observing six months of production for a real-world microcontroller by STMicroelectronics.**

***Index Terms*—Scan test, test economics, layout, ATPG.**

I. Introduction

OVER the past few decades, the automotive industry has witnessed an exponential rise in the number and intricacy of Automotive Systems-on-Chip (SoCs). This surge is driven by the massive demand for advanced functionalities in next-generation vehicles, coupled with a constant pressure to expedite time-to-market [1]. Technological advancements allow millions of gates to be integrated into single chips, enabling various and complex functionalities.

Received 2 May 2024; revised 28 November 2024; accepted 17 December 2024. Date of publication 23 December 2024; date of current version 13 March 2025. This work was supported by the Italian Ministry of Research and University under Grant DM1061. Recommended for acceptance by J. Li. *(Corresponding author: Giusy Iaria.)*

Giusy Iaria, Paolo Bernardi, and Lorenzo Cardone are with the Dipartimento di Automatica e Informatica, Politecnico di Torino, 10129 Torino, Italy (e-mail: giusy.iaria@polito.it; paolo.bernardi@polito.it; lorenzo.cardone@polito.it).

Claudia Bertani, Giuseppe Garozzo, and Vincenzo Tancorre are with the STMicroelectronics, 20864 Agrate Brianza, Italy (e-mail: claudia.bertani@st.com; giuseppe.garozzo@st.com; vincenzo.tancorre@st.com).

Digital Object Identifier 10.1109/TC.2024.3521246

Moreover, these significant improvements have also increased the complexity of the testing process, particularly for safety-critical applications. Proper device testing is crucial to avoiding life-threatening incidents. In these fields, such as Automotive, devices must guarantee high and reliable standards. For this reason, devices must undergo to various testing phases. The traditional Manufacturing Test Flow first consists of the Wafer Sort phase, where wafers, composed of all the dies, are tested. Then, the dies targeted as good are packaged and brought to the subsequent test flow step, the Package Test, in which the same pattern set of the Wafer Sort is typically applied, with the difference that the device is in its package. The test process cost is, therefore, composed of the cost required for the Wafer Sort, the cost of the packaging for the good devices, and the cost needed for the Package Test. For Automotive devices, the test flow continues with other subsequent steps, that are not the object of the paper.

All the test phases require substantial resource allocation in equipment, prompting chip manufacturers to actively seek methods for minimizing the number of devices undergoing the entire testing process [2], [3], i.e., by *shifting left* as much as possible the detection of faulty devices during the early testing stages, especially along the Wafer Sort test step. In case of detection, the scan test flow can have two modes: Stop-on-Fail and Continue-on-Fail. The Stop-on-Fail method quits the application of patterns as soon as a fault is detected, while the Continue-on-Fail doesn't stop the pattern application even if a fault has already been detected. Typically, the Stop-on-Fail flow is the most used in productive, industrial flows because it is less expensive in terms of test time and permits a certain reduction of costs for every failed device, especially in the case of testing every chip in single-site. Meanwhile, the Continue-on-Fail is helpful for the diagnosis process or collection of productive data but demands more testing time to apply all patterns to all dies. For instance, the Continue-on-Fail strategy is used to extract effectivity data from a sacrificial set of chips belonging to production lots devised for production monitoring.

Early identification of faulty devices on the production line is crucial for cost savings. Nonetheless, achieving high fault coverage at the Wafer Sort presents challenges, necessitating extensive time, memory, and monetary investments [4]. Automatic Test Pattern Generator (ATPG) tools generate many patterns for large and complex devices. The increase in the

pattern set's size directly affects the test time per device and, consequently, the scan test cost.

Many studies have attempted to solve the problem of the increasing number of tests to be applied. For example, in order to reduce the testing time, the solution proposed by several studies [5], [6] is reordering the test patterns to first apply the most effective ones. In this way, considering a Stop-on-Fail test flow, the testing time would be reduced in case of failure. Adapting these studies to large industrial chips costs lot of computational efforts, and reordering patterns according to their effectiveness could not always lead to a significant economic gain. In particular, if the used testing environment is multi-site [7], the gain is lower because multiple dies are tested in parallel, and good and failed devices could be tested simultaneously, deleting the eventual economic advantage of detecting *in less time* the failed ones. Other studies [8], [9], [10] have investigated a way to adaptively predict the number of useless patterns, thus maintaining only the useful ones or privileging the best ones to be executed earlier. Such approaches can be adopted also in multi-site environments.

Against this background, this paper aims to model the scan test costs such that it inputs Package Test and packaging costs in the overall computation, too. The proposed scan test cost model permits the computation of the economic trends that can be obtained, in different conditions of Yield, Fault Coverage, and test equipment costs per second, by considering cutting patterns from the tail of the complete pattern set as a *straightforward* alternative to pattern selection or reordering at Wafer Sort. It is shown that, in normal productive conditions, it is easy to gain in the overall scan cost by removing the latest patterns at Wafer Sort and accepting more expenses for the Package test based on the fact that ATPG tools, at the end of the generation process, always cover fewer and fewer faults than the patterns before [10].

The proposed model effectively and accurately determines how many patterns can be removed from Wafer Sort while still reaching a benefit even considering the extra cost along the Package Test. The proposed solution requires low computational and analysis effort and assumes that the complete pattern set is fully applied during the Package Test. This ensures that, at the end of the process, the number of good devices that exit the entire flow is the same as in the traditional approach.

The proposed model, therefore, allows us to estimate an ideal point to cut the patterns by starting from the end of the pattern set, where, while allowing more defective chips to pass Wafer Sort, it is still possible to achieve overall savings due to the reduction in Wafer Sort test time.

Reducing the patterns from the tail is an effective choice for a population of devices showing a uniform failure distribution. Nevertheless, in very large case studies the circuit may have areas with higher criticality than others, and in general, may have a *non-uniform failure distribution*, because some portions or gates of the circuit are more susceptible to possible systematic or random defects. For this reason, the economic gain achievable when cutting patterns generated with classical ATPG strategies, in which each fault has the same priority, could not be optimal.

To further refine the scan test cost model, this study proposes to input additional information about the potential circuit criticalities that may systematically or randomly manifest during manufacturing test steps of volume production of dies. The paper illustrates how to compute a Weighted Fault Coverage (Ω) value to input to the cost model to consider a non-uniform distribution of failures along production. Differently from other approaches [9], [10] that require continuous production monitoring to implement adaptiveness to a specific defectivity, one-shot computation of Weighted Fault Coverage Ω is performed over a sacrificial population of dies which is tested according to a Continue-on-Fail strategy, while Stop-on-Fail is used to the volume of produced dies.

In this context, as a by-product, the approach also suggests a methodology that guarantees that the first generated patterns cover the most critical areas of the circuit, when dealing with devices having non-uniform failure distribution. The ATPG generation is guided using a fault criticality rank to tackle more probable faults first, then maximizing the economic advantage of cutting patterns out from the tail of the patterns set.

Thus, to summarize, the proposed method is based on three key innovative elements:

1) based on a mathematical cost model, it propose to *shift to the right*, from Wafer Sort to Package Test, more devices than usual while reaching an economical gain, assuming that the full set of patterns is applied at Package Test, leaving the overall Yield unvaried;
2) including a combined analysis of failure data coming from a sacrificial population of devices tested with a Continue-on-Fail strategy and logic diagnosis data, it shows that it is possible to further refine the accuracy in the estimation of the test costs, thus enabling the identification of the optimal number of patterns to remove starting from the tail of any pattern set to achieve the largest economic advantage.
3) it illustrates that a proper test pattern generation guided by the aforementioned analysis to *shift to the left* along Wafer Sort the detection of those devices with faults laying in critical areas, hence maximizing the economic advantage reachable by cutting patterns from the tail of the generated set.

To achieve this objective, the study comprises at least three innovative contributions:

1) a **scan test cost model** to predict the ideal point to cut patterns from the tail of the pattern set for devices population presenting a non-uniform failure distribution. It elaborates on [11] and [12] formulas to also consider the mass defectivity distribution;
2) a **data analysis method** to determine the criticality level of the possible faults affecting the device, which is obtained from the analysis of failures resulted from a sacrificial lot of devices tested with the Continue-on-Fail test flow;
3) a **criticality-oriented ATPG algorithm** to generate patterns targeting the faults labeled as the most critical in the analysis phase, first.

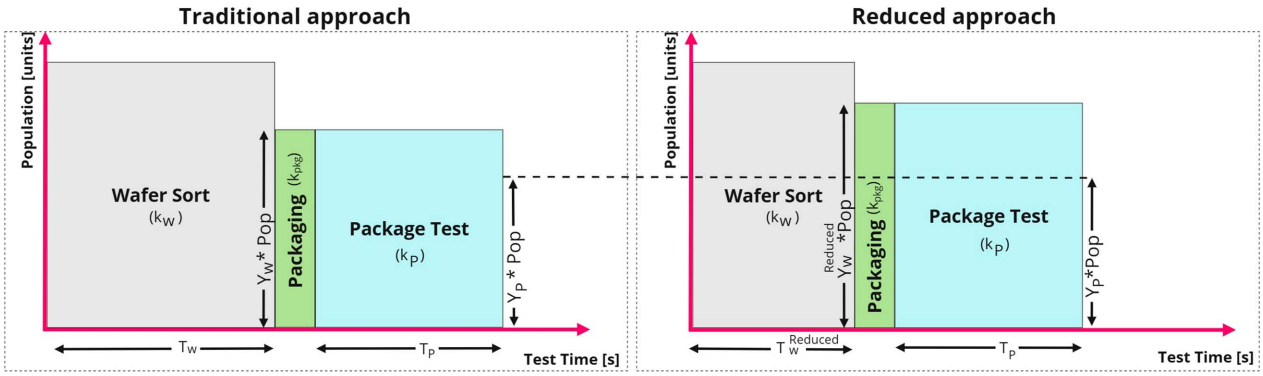

Fig. 1. Comparison at-a-glance between the traditional manufacturing test flow and the reduced approach.

The following sections provide detailed explanations of these contributions. Section II provides technical background about the addressed issue and references from state-of-the-art literature. Section III explains the proposed approach in detail. Section IV presents the experimental results. The approach is validated for an academic circuit benchmark with about 4 million Stuck-at and Transition delay faults and for a real-world Automotive $40nm$ SoC manufactured by STMicroelectronics, accounting for 80 million Stuck-at and Transition delay faults.

The results for the academic benchmark are distributed over the Section II and Section III as an example intended to guide the understanding of the described methodologies. The proposed model is finally used in Section IV to evaluate the cost trade-off for a large industrial chip tested with a pattern set encompassing hundreds of thousands of scan patterns. The circuit density is considered as a critical factor that weighs on the failure distribution over the production and used to compute the Ω over a population of failing devices. Theoretical plots of the scan test cost curve obtained using the proposed model are compared with data from about six months of volume production showing an extremely high correlation. The proposed model is also compared in section Section IV to the one presented in [10], showing better accuracy estimation of cost-cutting. Section V concludes the paper with some remarks on the key findings of the study.

II. Background

This section aims to provide the reader with the concepts that have driven this study and are necessary for its understanding.

A. Manufacturing Test Flow

The Manufacturing Test Flow consists of several phases applied one after the other to discard faulty devices [4], [13]. The same scan test set is repeated at various levels.

The first phase of the flow is the Wafer Sort phase. It checks for the primary electrical functionalities of the chip. At this level, dies are not yet packaged and the entire wafer is tested, usually resorting to scan patterns and possibly enabling multi-site testing of many chips at the same time [7]. The Wafer Sort cost depends on the number of patterns to apply to each device. Indeed the cost of the equipment is calculated on the time of use of the latter. As a consequence, a reduced pattern set would lead to a cost reduction for this phase.

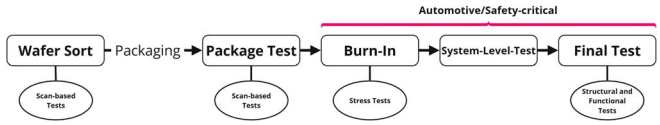

Fig. 2. Example of a possible manufacturing test flow.

After such a phase, the wafers are baked at high temperatures and, subsequently, each die is packaged and brought to the next phase, called Package Test. The Package Test is then performed on the dies that have been targeted as *good* by the Wafer Sort. It measures and tests the essential electrical characteristics of the pins and verifies that any issue is introduced along the packaging process. The cost of the Package Test is composed of the cost needed for the packaging of good dies and depends on the time of use of the equipment to apply the pattern set (just as for the Wafer Sort). Thus, the total cost here also depends on the goodness of the previous phase. Indeed, the more Wafer Sort escapes there are, the more the Package Test cost would be increased.

It is a common industrial practice to repeat the same scan-based test set used along Wafer Sort just after having packed the dies targeted as good. It should be noted that for safety-critical fields, the process continues with other steps with the aim of stressing the device and testing its functionalities in order to ensure the highest reliability level, as Fig. 2 shows.

The proposed strategy focuses on Wafer Sort and Package Test phases, intending to minimize their cost by reducing the number of patterns to apply at Wafer Sort while still sustaining the cost of shifting right some *extra* devices (the Wafer Sort escapes) to the Package Test.

B. Manufacturing Test Time Reduction Model

A straightforward and effective way to reduce the cost of scan tests is to reduce the test time by reducing the number of patterns in one of the phases of the entire flow. Reducing the number of patterns also leads to more test escapes in the subsequent steps. This paper focuses on the situation where the Wafer Sort is the object of the test time and cost reduction;

during the Package Test phase, the complete pattern set is fully applied again, as usual in production.

Fig. 1 aims to show graphically the joined costs of the Wafer Sort and Package Test phases, including packaging costs, with two different approaches, through the area occupied on the plot. The scan test cost is affected by three main factors: the size of the population, the test time for each phase of the testing process, and fixed costs that vary depending on the process itself. k_W is the fixed cost per second along the Wafer Sort, k_{pkg} is fixed the cost of the packaging per die, and k_P is fixed the cost per second along the Package Test. T_W is the test time required for Wafer Sort, T_P the one for Package Test. In the right side of the Fig. 1 there are also the terms $T_W^{Reduced}$ and $Y_W^{Reduced}$, representing respectively the reduced time along Wafer Sort and the consequent variation yield variation. The calculation of such terms will be detailed in the following formulas. To estimate the cost of a single die, the following equations can be used.

$$Cost_W^{Traditional} = k_W * T_W \tag{1}$$

$$Cost_P^{Traditional} = Y_W * (k_{pkg} + T_P * k_P) \tag{2}$$

The first formula estimates the Wafer Sort scan test cost by multiplying the time required by the test by a constant depending on the testing equipment cost. The second equation consists of the multiplication of two terms: Y_W, which represents the yield of the first testing phase, and the sum between the packaging cost (k_{pkg}) and the result of the product between the time required for Package Test (T_P) and the operating cost of the latter (k_P). By multiplying the sum of the two costs by the size of the initial population (Pop), it is possible to compute the cost for all the population.

$$Cost_{Tot}^{Traditional} = Pop * \left(Cost_W^{Traditional} + Cost_P^{Traditional}\right) \tag{3}$$

To model the reduction of the Wafer Sort, the following formula, that also describes the second half of the Fig. 1 can be used:

$$Cost_W^{Reduced} = k_W * T_W^{Reduced} \tag{4}$$

In the previous equation, the time factor for the calculation of $Cost_W^{Traditional}$ is replaced with a generic *reduced time*. In order to reduce said time, applied patterns are cut by discarding the last ones. Other works [10] already discussed the effect that these patterns have on the ability of the test to catch a faulty chip. Because of the reduction of the number of patterns, the accuracy of the test will decrease, increasing the yield, since some defective devices will no longer be screened. The new yield ($Y_W^{Reduced}$) is the sum between the yield that would result from the classical approach (Y_W) and an additional term representing the test escapes (TE) resulting from the Wafer Sort time reduction. The latter is estimated with the known formula [11], where FC is the function representing the fault coverage vs the number of applied patterns (x).

$$TE = 1 - (Y_W)^{1-(FC(x)/100)} \tag{5}$$

$$\begin{aligned} Y_W^{Reduced} &= Y_W + TE \\ &= Y_W + 1 - (Y_W)^{1-(FC(x)/100)} \end{aligned} \tag{6}$$

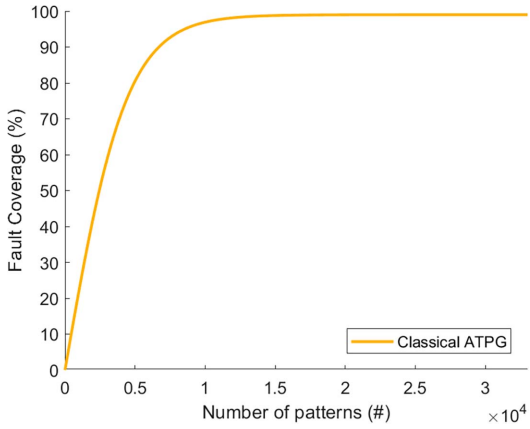

Fig. 3. Fault Coverage vs percentage of applied patterns.

By substituting this new yield in the previous equation of the cost of Package Test, it changes as follows.

$$Cost_P^{Reduced} = Y_W^{Reduced} * (k_{pkg} + T_P * k_P) \tag{7}$$

$$Cost_{Tot}^{Reduced} = Pop * \left(Cost_W^{Reduced} + Cost_P^{Reduced}\right) \tag{8}$$

The total cost (Equation 8) of the reduced approach directly depends on the reduced time of the Wafer Sort ($T_W^{reduced}$) and on the yield ($Y_W^{Reduced}$). Thus, once fixed the constant costs of the testing environments (k_W, k_{pkg} and k_P), Equation 8 models the behavior of the cost changing depending on the time reduction and consequent increase of the test escapes, as stated in Equation 6.

Since the $Cost_W^{Reduced}$ increases with the time of testing, and the $Cost_P^{Reduced}$ decreases with lower yield of the first step, a minimum in the cost function can be found. To better explain the behavior of the costs an example with an academic circuit benchmark is provided.

Example 1 ■

This example includes consideration about an academic benchmark, that is a cluster of 24 OpenRISC1200 processors [14] showing about 2 million Stuck-at and 2 million Transition delay faults.

An exhaustive ATPG pattern generation has been performed. The result was a pattern set composed of about 30 thousand patterns reaching 99% of fault coverage. The trend of the fault coverage vs the number of patterns is shown in Fig. 3.

Meanwhile, Figs. 4 and 5 show the estimated costs, using the formulas previously described, of the testing procedures with different values of yield, Wafer Sort and Package Test costs. In particular, two hypotheses for the yield are considered: a high yield, with a value of 99%, and a low yield with a value of 90%.

If considering further lower values of yield, like 80%, the achievable economic gain for the testing process is even higher. In the case of this example, the maximum achievable gain would be ≈ 6.3% with 80% yield. Nevertheless, it's not very common in a mature production to see such low yield numbers unless during crises that have limited durations. Low yields should be managed from a technological point of view and less from

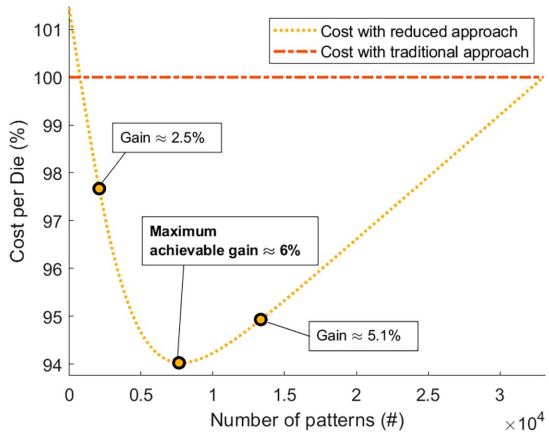

Fig. 4. Low yield (90%), $k_W/k_P \approx 0.15$.

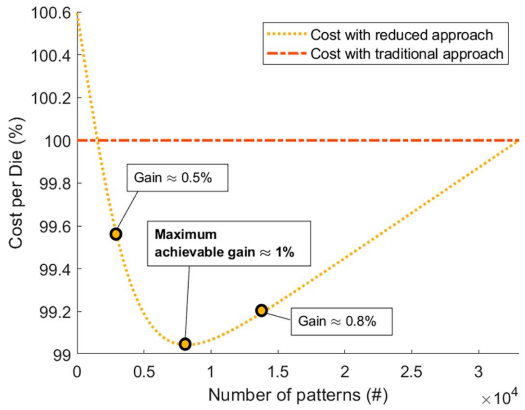

Fig. 5. High yield (99%), $k_W/k_P \approx 0.018$.

a testing perspective, which is the main topic of the proposed model.

Moreover, different values of fixed costs for Wafer Sort and Package Test costs are valued, with two different ratios (k_W/k_P): 0.15, and 0.018. The packaging cost (k_{pkg}) is the same in both hypotheses. The location of the minimum depends on the considered costs and yield. The higher k_W is, the more influential the cost of the Wafer Sort is, and the steeper the curve rises after the minimum, since the time required for this step is increased. The higher the sum $k_{pkg} + k_P$ is and the sharper is the decrease in cost in the first section of the curve, since less faulty devices are escaping the Wafer Sort procedure.

Using the described model, it is possible to find the maximum achievable gain for both cases. For Fig. 4 the maximum gain, which corresponds to the minimum point of the curve, is around 6%. Meanwhile, for Fig. 5 is about 1%. Precisely, even if they may look similar at first glance, they have very different cost savings at the knee of the curve. Fig. 4 has a higher Wafer Sort cost, which, if reduced, can achieve a significant cost reduction. On the contrary, Fig. 5 is characterized by low Wafer Sort cost that does not produce the same behavior. Nonetheless, the high yield allows for a sharp reduction in the number of patterns before the cost of the faulty chips escaping the Wafer Sort impacts the packaging and Package Test costs. The steeper the curve, the higher the achievable gain. In addition to the minimum point, two more points are pictured in the surroundings, which are about $5,000$ patterns distant from it. In Fig. 4, if the cut is too severe, the gain decreases til 2.5%; if it is too moderate, the gain decreases til 5.1%. A similar behavior happens in the other case (Fig. 5) but with smaller values. Thus, it is crucial to correctly estimate the best point to cut patterns to get the maximum achievable gain.

End of the example 1. ■

C. State-of-the-Art About Automatic Test Pattern Generator and Pattern Reduction Methodologies

Automatic Test Pattern Generator (ATPG) tools are tasked with generating patterns to test digital devices, trying to cover Device Under Test (DUT) faults as much as possible. However, increasing complexity, in terms of logic gates in modern automotive SoCs, has brought also complexity to the process of generating test vectors. Even considering the terms of constraints and computation time [15].

If there are no particular generation guidance, the fault coverage obtained by generating a pattern through ATPG is usually incremental in the first few patterns, even fast, depending on the case study. However, at the same time, it tends to stabilize or increase slowly after several test patterns. For this reason, an incremental approach is often used [16], [17] to divide the circuit into small portions to target and to cover multiple fault models.

Among the problems slowing the growth of fault coverage for ATPG-generated patterns is the increasing complexity of modern devices, with huge number of logic gates and flip-flops. Iterative incremental ATPG executions are required to reach high test coverage and fulfill safety standards, which directly affects the size of the pattern set.

As the complexity of automotive SoCs increases, topology information about the placement of standard cells is beginning to play a crucial role in the development and analysis of these systems. In recent years, researchers have proposed different approaches to analyzing and exploiting layout information. For example, [18], [19], [20] presents a new methodology for extracting bridge faults from calculating critical areas based on layout information is described. The resulting list of bridge faults is also sorted by critical area size. Then, eventually, the list can be truncated to give precedence to bridge faults characterized by a larger area. The authors presented experimental results on an Automotive $160nm$ case study. The results show that patterns generated by targeting bridge faults extracted in this way actually add coverage compared to patterns generated by traditional methods, even in the case of CAT (Cell-Aware Test). The study also analyzes how much the extracted list can be reduced while still maintaining an increase over traditional coverage. The patterns generated for bridge faults are in addition to those generated typically, so the method does not aim for a test cost reduction but more to add accuracy in the testing process. Also in [21] a study is proposed about targeting only faults in denser areas for pattern generation.

The information on the layout phase could be exploited to drive better the test flow efforts. Given the ever-increasing increase in testing costs, there are several studies in the literature that attempt to lower these costs with various methodologies. For example, [8] aims at reducing the patterns to be applied for each core of the device during Wafer Sort. The strategy adopted is based on the combination of statistical yield modeling and integer linear programming. The study aims to determine the best number of patterns to apply for each device core in case there are test time constraints during wafer sort. Experimental results show that screening during wafer sort remains high using the proposed sorting technique. The case studies used are 5 of the ITC'02 benchmarks [22].

Another way to reduce the test cost is reordering the patterns moving forward the ones with higher failure rate, as described in [5]. The approach uses a SVMRANK algorithm to find the optimal pattern sequence. The reported case studies are some benchmarks from [23], so the experimental results are made on simulated experiments.

In [6], the authors propose an adaptive testing technique that is based on the results obtained from the previous tested die. Considering multi-site wafer testing environment, this technique could be hard to adopt. Simulation results are reported for a benchmark from [23].

In addition to this, the study described in [10] discusses on the number of patterns that can be actually useful during the testing process. The study proposes a statistical model with the goal of finding the number of useful patterns to apply and thus to reduce the final test cost. Such a model is cost-independent but depends on the number of tested devices. In particular, it is based on the function $F_D(k)$, that is defined as the probability of detecting a defective chip when patterns 1 through k are applied. The function is extrapolated by production data and the study states that it can be easily adapted depending on the case study. The latter will be compared to the proposed methodology to find the best point to cut the patterns.

The proposed methodology differs from the other ones because it proposes a cost model for finding the ideal point to cut patterns from the tail, even in non-uniform failure distribution and considering the overall cost of Wafer Sort and Package Test. As an additional by-product, it proposes a new pattern generation strategy so that the most critical faults are covered first. Moreover, the proposed methodology reduces the number of patterns for all the tested devices, even for the good ones. Meanwhile, the studies focusing on reordering patterns actually reduce patterns just for the failed devices.

III. The Proposed Approach

The proposed methodology provides a comprehensive and effective scan test cost model capable of enabling optimization of the scan test cost considering two consecutive test flow steps, such as Wafer Sort and Package Test.

The proposed approach bases on the assumption that in productive environments scan-based tests applied at Wafer Sort are then repeated after the device packaging, along the Package Test phase. In this context, as discussed in Subsection II-B, reducing the cost of scan tests can be possible by cutting patterns to apply from the tail of the pattern set. The described formulas help finding the best point to cut patterns during the Wafer Sort and to still have an economic gain despite more test escapes sent to the Package Test after the Wafer Sort, where they are finally screened out. Such formulas are valid for any pattern set and apply when the failure distribution is uniform. Adapting those formulas to real industrial scenarios is the main object of investigation. Thus, the cost model is extended, exploiting the study reported in [12], to be effective also for real-world devices with complex designs, which may have areas with more faults likely to occur than others.

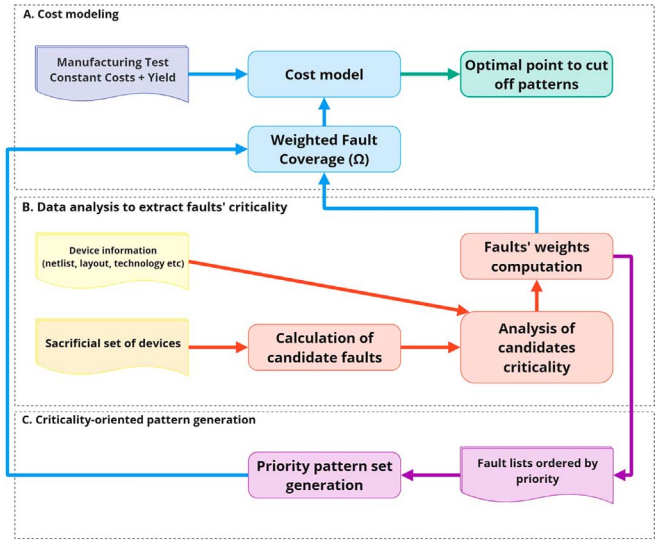

Fig. 6. Brief workflow of the proposed approach.

Hence, a comprehensive and extended scan test cost model is proposed. The latter is capable of predicting the cost reduction in the case of non-uniform failure distribution. Such model extends the concept of Fault Coverage to *Weighted Fault Coverage* (Ω) [12], which takes in consideration also the probability for each fault to occur.

This work aims to optimize the economic gain achievable by reducing the number of patterns. An effective pattern generation strategy is proposed to maximize the gain. The outcome is a final pattern set that fits better with the circuit criticality, bringing a higher economic gain than a classical application flow would.

Fig. 6 shows briefly the steps of the proposed methodology, in which three branches can be distinguished:

A) *Cost Modeling for non-uniform failure distribution*: by modeling the scan test cost, an ideal point for cutting off patterns is estimated, where, while increasing Wafer Sort escapes, it is still possible to achieve overall cost savings due to the reduction in Wafer Sort test time. The model uses the Weighted Fault Coverage (Ω), which involves the probability for each fault to occur, making the methodology suitable for devices presenting non-uniform failure distribution;

B) *Data analysis for the extraction of the criticality level of faults*: by analyzing failures testing with the Continue-on-Fail flow a sacrificial batch of devices, the faults more prone to be candidate are found, and each fault is assigned with a criticality index (weight). The computed weights enable the calculation of the Weighted Fault Coverage (Ω), used in the cost model. The latter exploits the weights to guarantee an accurate prediction in case of non-uniform failure distribution;

C) *Criticality-oriented ATPG for pattern set generation*: by exploiting the analysis of the step B, in which faults are ranked depending on a criticality index, a new pattern set is generated following the priority of the ranking.

A. Cost Modeling for Non-Uniform Failure Distribution

The proposed cost model, for non-uniform failure distribution, extends the formulas described in Subsection II-B.

In particular, Equation 6, describing the $Yield_W^{Reduced}$, is based on the well-known formula proposed by [11], in which uniform distribution is assumed.

Equation 6 ($Yield_W^{Reduced}$) takes into account the test escapes (Equation 5) introduced by the reduction of test time along Wafer Sort. Such term is significant because it describes the increase of the Package Test cost due to the increase number of test escapes coming from Wafer Sort. Equation 6 is based on the well-known formula proposed by [11], in which uniform distribution is assumed. The latter formula depends on the Fault Coverage. Considering non-uniform distribution, the formula for the yield changes taking into account also the probability for a fault to occur, following the derivation process described in [12]. Thus, the new formula that describes the yield resulting from a reduced application of Wafer Sort, for non-uniform failure distribution is:

$$Y_W^{Reduced} = Y_W + 1 - (Y_W)^{1-(\Omega/100)} \tag{9}$$

where Ω represents a Weighted Fault Coverage that also considers the probability for each fault to occur, contrary to the classical Fault Coverage metric that assumes each fault to share the same weight. The formula to compute such value is shown in Equation 10, that is a weighted average.

$$\Omega = \frac{\sum_{n=1}^{N} W_n \cdot FD_n}{\sum_{n=1}^{N} W_n}, \text{ where } FD_n \in \{0,1\} \tag{10}$$

N represents the total number of faults. The numerator of the division is composed of the weighted summation of all the detected faults. Each fault is weighted depending on the W_n value ($W_n \in [0,1]$), and each weight is multiplied by a binary variable (FD_n), which represents whether that fault was detected (1) or not detected (0) by the considered pattern. Repeating the computation for each pattern (or chunks of patterns) gives the trend vs the number of applied patterns.

The Weighted Fault Coverage (Ω) enhances the accuracy of predicting how many test escapes would occur for devices presenting a non-uniform failure distribution. The prediction of the number of test escapes directly affects the Package Test cost. Since the aim of the paper is finding the optimal point where reducing Wafer Sort test time leads to an economic gain despite the increase of test escapes and Package Test cost. Thus, the more accurate the prediction of test escapes, the more accurate the cost gain estimation.

An effective way how to compute the Ω in real production cases is detailed in the following subsection.

B. Data Analysis to Extract the Criticality Level of Chip'S Gates

In order to extract the criticality of the faults (and to compute thus the Ω), a strategy is proposed which encompasses the following elements:

- A corner lot of sacrificing devices tested with a Continue-on-Fail flow, applying the pattern set used in production.
- A ranked list of the total candidate faults computed by considering the number of occurrences of each candidate per each failing die.
- A list of criticality classes extracted through the device information (netlist, layout, technology) to extract the correlation between the extracted candidate faults and their criticality index, in order to compute the weight of each fault and finally the Ω.

The criticality indexes could diverge depending on the used technology. Thus, in order to make the methodology independent from the latter, the use of a sacrificial batch of devices can be exploited to tune the critical areas for all the devices using the same technology, i.e. belonging to the same chip family.

The key elements of the Ω calculation are: 1) computation of candidate faults per each failing die, 2) criticality analysis of such candidates.

The single-fault hypothesis enables considering as candidates only those faults covered by all failed pattern sets, thus by the intersection of their coverage sets, allowing for removing all faults covered by the pattern sets that did not fail. Considering F as the total number of failed pattern sets, and the set of covered faults for each failing pattern set ($Fail^{per\_pat}$), the intersection of all the failing pattern sets is the starting point of the candidates calculation:

$$Fail_{die} = \bigcap_{f=1}^{F} Fail_f^{per\_pat} \tag{11}$$

$Fail_{die}$ represents the intersection between all the failing patterns. At this point, the set can be reduced further, considering also the succeeded pattern sets. Indeed, the faults covered by succeeded patterns cannot be candidates.

Considering S as the total number of succeeded pattern sets, and the set of covered faults for each successful pattern set ($Success^{per\_pat}$), the union of all the sets represents the faults that cannot be candidate and that must be deleted by the $Fail_{die}$ set. Thus, the final set representing the candidate faults for a die is Equation 13.

$$Success_{die} = \bigcup_{s=1}^{S} Success_s^{per\_pat} \tag{12}$$

$$Candidates_{die} = Fail_{die} - Success_{die} \tag{13}$$

Algorithm 1 Faults' weight computation

Input: Population of faults ($fault\_list$), Ranked candidate faults with the number of occurrences of each ($ranked\_candidates$), Criticality classes with the faults belonging to them ($criticality\_classes$)
Output: Weighted fault list ($weighted\_fault\_list$)

$weighted\_fault\_list \leftarrow [\emptyset]$
$N\_classes \leftarrow len(criticality\_classes)$
$classes\_freq \leftarrow N\_classes * [0]$ ▷ it keeps count of the candidates' occurrences for each criticality class
for each $fault$, $occurrences$ in $ranked\_candidates$ **do**
    $class \leftarrow criticality\_classes$.find($fault$)
    $classes\_freq[class] \leftarrow classes\_freq[class] + 1$
end for
$ranked\_classes \leftarrow$ get_ordered_classes($classes\_freq$) ▷ ranked classes by the candidates' occurrences
for each $fault$ in $fault\_list$ **do**
    $class \leftarrow criticality\_classes$.find($fault$)
    $rank \leftarrow ranked\_classes$.find($class$)
    $fault.weight \leftarrow \frac{N\_classes - rank + 1}{N\_classes}$
    Append $fault$ to $weighted\_fault\_list$
end for

Once the candidate faults are extracted, they are ranked depending on how many dies have that fault in their candidate list. Such ranking is then exploited by Algorithm 1 to cross the information about the criticality levels.

To extract the criticality of a device, its information, such as netlist, layout and technology, could be exploited. Different studies have proposed for example to exploit the layout information. In [18] a study is presented about increasing the total coverage in production exploiting critical areas of the layout. Moreover, [21] and [24] discussed how in modern complex devices the layout of the circuit is characterized by non-uniform distribution of gates. Such studies proposed hence a way to correlate the density and the device criticality.

Nevertheless, the way in which the criticality classes are extracted is not relevant. Indeed, once had the latter, the computation of the weights for the faults and the Ω can be performed independently from the extraction. The only requirement is to have criticality classes together with the faults which belong to each class. The proposed algorithm to compute the weights is detailed in Algorithm 1. Such weights are then exploited to finally get the Weighted Fault Coverage (Ω) as shown in Equation 10.

Example 2 ■

Continuing the example made in Subsection II-B, the same academic circuit benchmark is used to show the calculation of the Weighted Fault Coverage (Ω).

Since it is not possible to have a real sacrificial set of dies for the benchmark, multiple populations of about $1M$ dies with a high yield (see Fig. 5) have been created, by selecting the faults from the fault universe, in Montecarlo fashion [25]. The selection of the faults has not been made solely randomly, because a non-uniform failure distribution is considered.

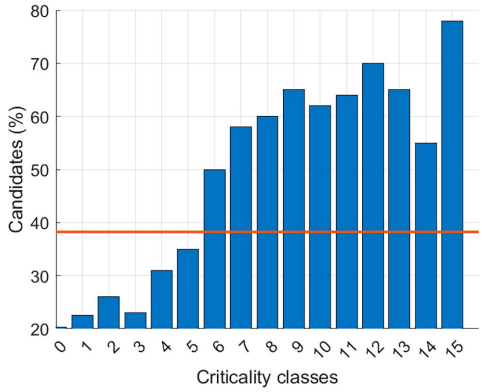

Fig. 7. Percentage of candidates for each criticality class (number of neighbors).

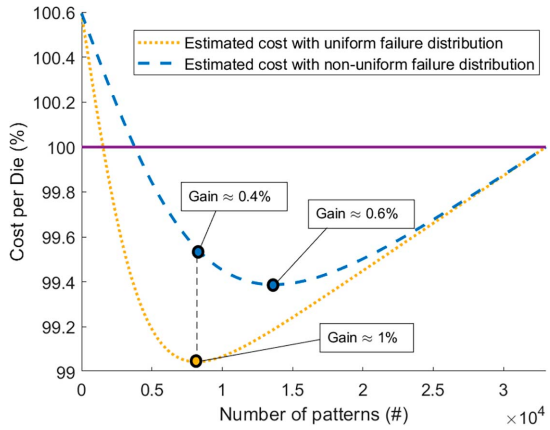

Fig. 8. Cost per Die (%) vs number of applied patterns considering both uniform and non-uniform failure distribution.

The circuit has been synthesized, and the place and route phase has been completed with the same tool and technology ($40nm$) used for the industrial case study, that will be reported in Section IV. As it will be seen in such a section, the criticality index chosen is the density metric [26].

In particular, when selecting the faults to simulate a real testing scenario, it was considered that the fault distribution is non-uniform. Consequently, the selection probability of a fault was made depending on its classification as belonging to a more or less critical class. Since the same technology is used for the academic benchmark, the same assumption can be made about the correlation between the density areas and the criticality. Fig. 7 shows how the candidates are distributed in average over the Montecarlo simulations. It provides a general understanding of candidates distribution in relation to gate density. The x axis represents the density classes of the device, and the bars identify the percentage of how many candidates were found in that specific class.

Once the faults' weight are computed following the Algorithm 1, the cost estimation for the non-uniform failure distribution can be made.

Fig. 8 shows the curve trend for the estimated cost with uniform failure distribution (the same situation shown in Fig. 5)

Algorithm 2 Criticality-oriented ATPG pattern generation

Input: List of criticality classes ordered by their priority ($criticality\_classes$), Fault Lists for each criticality classes ($fault\_lists$)
Output: criticality-oriented pattern set ($criticality\_pattern\_set$)

```
Run ATPG tool
criticality_pattern_set ← [∅]
for each class in criticality_classes: do
    Read faults from fault_lists[class^th]
    Create patterns
    Store patterns to criticality_pattern_set
end for
```

and also the curve with the cost estimation computed by the proposed model, which also considers the Ω calculation. The uniform model predicts a maximum achievable gain of about 1% cutting patterns at around $8,000$. Meanwhile, the non-uniform model predicts a maximum gain of about 0.6% around $13,500$ patterns.

The cost model considering uniform failure distribution is slightly more optimistic in predicting the maximum achievable gain. A similar behavior will be shown in Section IV for the real-world industrial case study, where both the estimations will be compared to real production data collected over six months. However, the best point to cut patterns predicted by the *uniform* model actually would bring to a gain of about 0.4% considering the *non-uniform* distribution, which is more accurate (as it will be shown in Section IV).

Example 3 will show how the ranking of the criticality class will be exploited to generate a criticality-oriented pattern set, that optimizes the maximum achievable economic gain.

End of the example 2. ■

Given any pattern set, the Ω enables to better estimate the economic cost gain compared to the uniform model. Moreover, the economic gain can be maximized basing on these information, as shown in the next subsection.

C. Criticality-Oriented ATPG for Pattern Set Generation

The result of the *analysis* phase will be a list of weighted faults, with each weight corresponding to the fault's criticality class. Therefore, such an information can be exploited to create a pattern generation driven by the criticality of the faults. Generation can be done by a traditional ATPG tool; however, the ATPG will be directed step by step to the faults deemed most critical to the device and then continue to cover all the gradually less critical faults until the list of faults is exhausted.

The goal is to cover the most critical faults first and leave the less critical ones at the bottom. In this way, by reducing patterns from the bottom, one can be assured that the faults left out are the least likely to occur. The algorithm used to implement this strategy called *criticality-oriented ATPG* is described in Algorithm 2. This generation strategy is a by-product of the approach. It aims to further maximize the economic gain

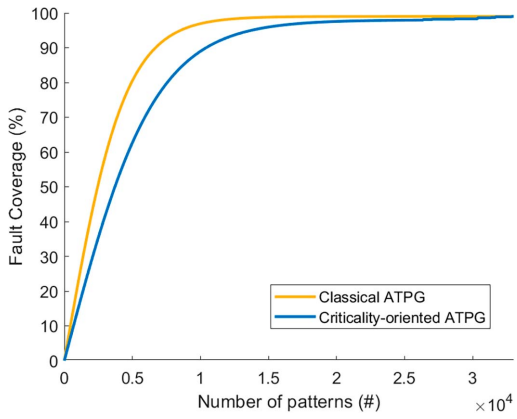

Fig. 9. Fault coverage (%) comparison between classical ATPG and criticality-oriented pattern generation.

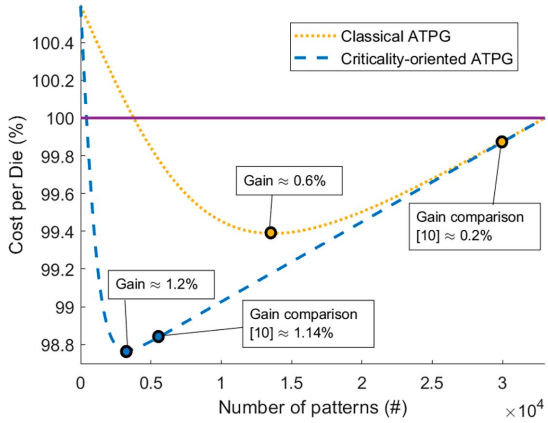

Fig. 10. Non-uniform failure distribution: Cost per Die (%) vs number of applied patterns generated by classical ATPG and criticality-oriented ATPG algorithm.

achievable by reducing patterns during Wafer Sort for devices presenting non-uniform failure distribution.

Example 3 ■

Referring to the academic benchmark considered so far, to better clarify the methodology through actual examples, a new pattern set has been generated following the Algorithm 2. The new pattern set has been generated with the same number of patterns of the traditional one (≈ 30 thousands), but guiding the generation as described in Algorithm 2. The comparison of the two pattern sets is shown in Fig. 9.

The trend of the criticality-oriented generation is a bit slower. This behaviour is reasonable because the required efforts for the guided ATPG are higher than the classical procedure. The reached coverage is a bit less than the original one, with the same number of patterns.

Fig. 10 shows the curves trend for two different pattern sets, computed by the proposed cost model considering non-uniform failure distribution, thus exploiting the previously computed Ω (see Example 2).

The model can predict which is the best point to cut patterns to achieve the maximum economic gain, also considering the

non-uniform failure distribution, for any given pattern set. Thus, the methodology can be applied just to know where to cut the pattern set already in use. Moreover, Fig. 10 also shows the points to cut patterns predicted by the model presented in [10]. They are reported in the non-uniform curves because they more accurately predict the production trend (as it will be shown in Section IV with real production data).

As a by-product of the analysis, the extrapolated information about the failure distribution can also be exploited to generate a more accurate pattern set, which follows the algorithm shown in Algorithm 2. The predicted maximum economic gain for the pattern set generated by classical strategy is about 0.6%. Meanwhile, exploiting the criticality information, the newly generated pattern set achieves double the gain, reaching 1.2%.

End of the example 3. ■

IV. Experimental Results

This section provides the reader with the experimental results obtained for all stages of the proposed methodology on the case studies. In particular, the results obtained for an academic circuit benchmark have been shown so far in the paper through the examples provided. Thus, in this section the results obtained for the industrial case study produced by STMicroelectronics are reported.

First of all, the experimental setup and the cost analysis on the pattern set currently used in production will be described. Then, by exploiting data collected for a sacrificial lot of devices, the failure distribution on the tested devices is analyzed and the possible faults are categorized according to their criticality class. The test cost analysis considering both uniform and non-uniform failure distribution will be discussed, comparing their cost prediction to real production data collected over six months.

Finally, the pattern set generated by the *criticality-oriented ATPG* algorithm is reported, showing how generating the patterns following the proposed method do optimize the test cost for devices with non-uniform failure distribution.

The comparison with the model proposed by [10] to find the number of useless patterns is also reported.

All the experiments regarding the pattern generation and the fault simulation have been executed on a high-performance multi-processor server equipped with a 64-bit 16-core processor AMD EPYC 7301, 256GB of RAM, a storage system of 10 TBytes, and a Centos Linux 7 operating system by using *TestKompress* from the Tessent-ATPG suite.

A. Experimental Setup

The considered industrial case study to validate the whole procedure is a large Automotive System-on-Chip produced by STMicroelectronics [21], [24], with the following characteristics:

- $40nm$ technology;
- about 20 million gates;
- about 700 thousand flip-flops;
- multi-core architecture;
- ASIL-D compliant.

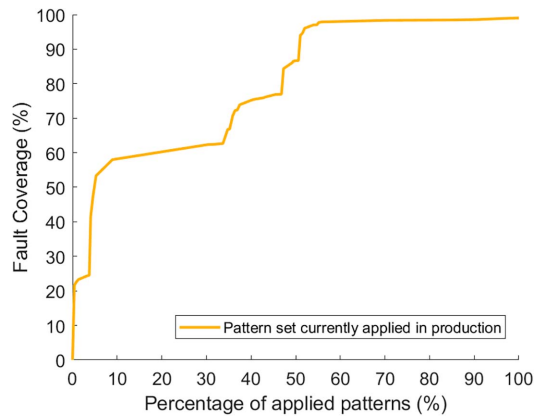

Fig. 11. Pattern set currently used in production: Fault coverage vs percentage of applied patterns, ordered by the sequence of application.

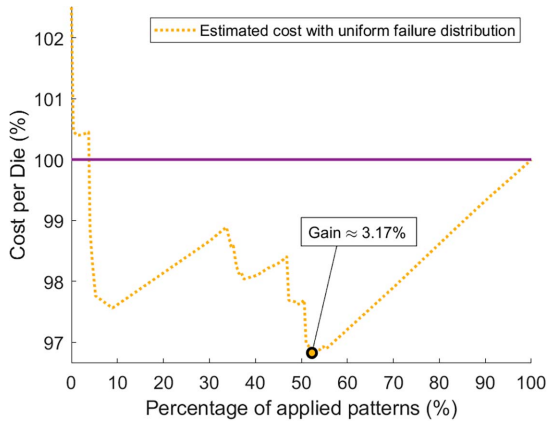

Fig. 12. Pattern set currently used in production: Estimated Cost per Die (%) considering uniform failure distribution vs Percentage of applied patterns (%).

The complete fault list size of the case of study is composed of about 80 million Stuck-at and Transition delay faults.

The case study supplied by STMicroelectronics includes also the industrial ATPG pattern set used during the production process. The patterns target both Stuck-at and Transition delay faults. Fig. 11 shows how the fault coverage of such a pattern set grows depending on the percentage of applied patterns. The patterns are provided and reordered by the company, so the trend does not follow the generation process by the ATPG tool. The pattern reordering results from a field evaluation process carried out during mass production according to consolidated industrial techniques that cannot be disclosed. Since the pattern set is currently applied with this order, it is used as the starting point of the analysis, with the aim to first show that even with a consolidated pattern set it is possible to estimate a significant gain by reducing patterns from the tail during Wafer Sort.

In Fig. 12 the trend cost, assuming uniform failure distribution, is pictured using the described formulas in Subsection II-B, with high production yield and high packaging and package test cost. Exact numbers cannot be disclosed and percentages are reported.

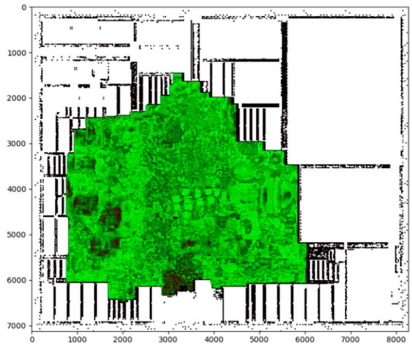

Fig. 13. Coloured gate-density distribution over the layout of a large Automotive System-on-chip.

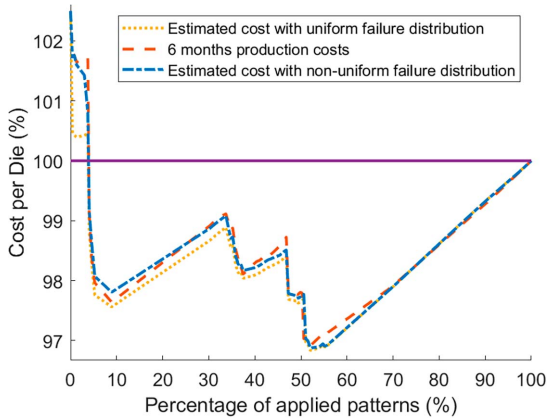

Fig. 14. Estimated costs and six months production cost vs percentage of applied patterns.

Using the cost model considering the uniform failure distribution, the estimated economic gain that would have been achieved reducing the pattern set currently used in production, only during Wafer Sort, is around 3.17% per device.

Moreover, in the next section, the data analysis of the failure distribution will be detailed. Such an analysis will then be exploited to estimate the cost trend using the proposed cost model, which also considers the non-uniform failure distribution. Both estimations will also be compared to real production data.

B. Faults' Weight Computation

The proposed cost model assumes a non-uniform failure distribution and exploits density as a factor. To know which faults are more prone to occur and to assign them a weight, depending on a criticality index, a sacrificial set of devices of a meaningful population has been tested using the Continue-on-Fail flow.

After having fault simulated each applied pattern set to retrieve the individual list of covered faults and the corresponding fault candidates, a tool has been developed in Rust language to implement the algorithm described in Algorithm 1. At the end of this process, the result is a ranked list of candidate faults of the failing dies. For the assignment of the weights the density metric (already discussed in [24], [26]) has been used. Thus, a *density class* is considered a criticality class. A density class specifically refers to the number of neighbors each gate has upon a certain distance, typically defined as the input pins distance of an AND gate [26].

The case study layout is shown in Fig. 13 as an example of non-uniform gate distribution in the front-end. The gate density on the physical layout is not uniformly distributed over the entire device surface.

The use of colors further highlights the density difference: a brighter shade of green describes parts with a higher gate density. Contrarily, a darker shade of green indicates zones in which fewer gates have been placed.

The faults are categorized by their density class and the such information are then crossed with the candidate faults resulted from the tested sacrificial lot. As described in subsection III-B, the weights of each fault are computed and used to finally calculate the Weighted Fault Coverage (Ω) that will be used in the next section to illustrate the proposed cost model.

C. Test Cost Analysis

An equation to estimate the cost of cutting patterns at Wafer Sort accepting extra costs for packaging and during Package Test was already introduced in Subsection II-B. However, that has been computed assuming a uniform failure distribution. Not all the faults are always equally probable, and, because of this, the real production data can deviate from the made estimation, that is solely based on the Fault Coverage.

Fig. 14 shows:

1) the cost trend modeled by the cost model assuming uniform failure distribution;
2) the real costs coming from six months of production;
3) the trend modeled by the proposed cost model considering non-uniform failure distribution.

All the curves depends on the number of patterns applied along Wafer Sort, since the aim of the paper is to show how cutting time during Wafer Sort could bring to an economic gain despite the increase of test escapes to Package Test. In the case of the real production data, the number of test escapes is not estimated but it is the real one.

As it can be seen, while it is true that the uniform failure distribution estimation do not match completely with the real production cost curve, they still share the same behavior. Zooming in the figure in the minimum point for the cost function (Fig. 15), it is possible to notice that the estimation of the non-uniform model, based on the Weighted Fault Coverage (Ω) is slightly more precise than the uniform model, solely based on the Fault Coverage. The points shown in Fig. 15 regard the actual gains that could have been achieved in production depending on the cut point. Fig. 15, showing a zoom in the area of minimum cost, highlights that a small mistake in cutting patterns along Wafer Sort could bring to a lower economic gain, even if the mistake is pretty small. It shows that the maximum economic gain is around 3.1% per device, around 52% of patterns applied. If the cut is around 49% of patterns, the economic gain decreases to 2.3%; if the cut is around 47%,

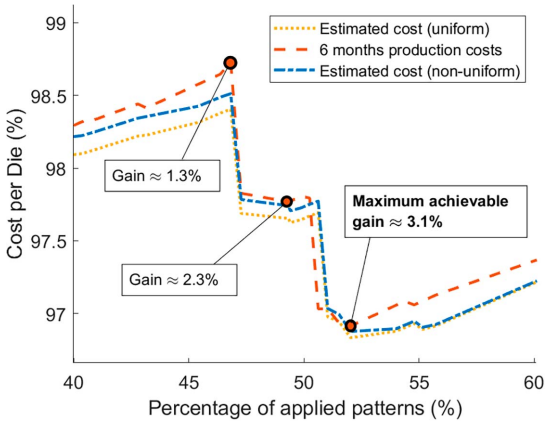

Fig. 15. Zoom over the region of minimum cost.

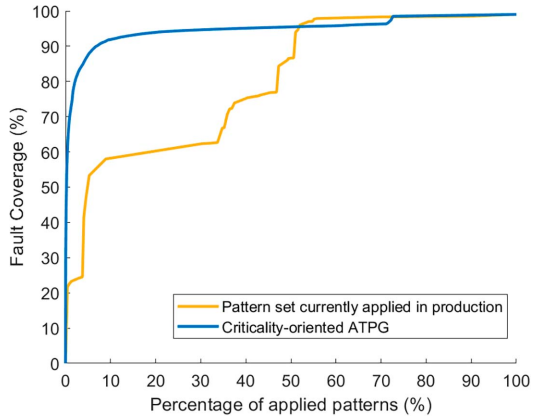

Fig. 16. Fault coverage vs Percentage of applied patterns.

TABLE I
ACCURACY OF THE MODELS IN RESPECT TO THE RESULTS COLLECTED ALONG SIX MONTHS OF PRODUCTION

Month [#]	**Pearson Correlation Index**		**Average Error [%]**	
	Uniform	**Non-Uniform**	Uniform	**Non-Uniform**
1	0.966	0.991	0.17	0.07
2	0.967	0.991	0.17	0.07
3	0.968	0.991	0.16	0.07
4	0.968	0.990	0.16	0.08
5	0.967	0.991	0.17	0.07
6	0.969	0.990	0.15	0.08
Cumulative	0.968	0.991	0.16	0.07

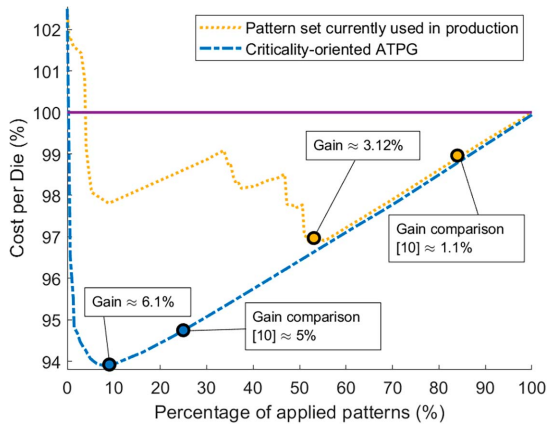

Fig. 17. Non-uniform failure distribution: Cost per Die (%) vs Percentage of applied patterns (%).

the economic gain decreases to 1.3%. Thus, if the cut happens too early, it could lead to a lower economic gain or, worse, to a cost increment. The same could happen if the cut happens too far. Predicting the optimal point where to cut the patterns is crucial to have the highest economic gain.

The maximum economic gain that would have been achieved by reducing the pattern set currently used in production, only during Wafer Sort, is around 3.1% per device, considering data collected in production over six months. The uniform cost model predicted the maximum economic gain to be 3.17%, and the proposed non-uniform model reaches a more accurate measure of 3.13%. The values are similar in the minimum point region, which is predictable because the pattern set has been generated without any specific guide. For this reason, the faults' coverage has been spread uniformly.

Table I shows the Pearson correlation index and the average percentage errors along six months of production for both uniform and non-uniform failure distribution model. The latter achieves 2% more correlation and it has a lower average error, looking at the cumulative values (0.07% vs 0.16%).

To further optimize the achievable economic gain, in the next section results for a new pattern set that also considers the non-uniform failure distribution are reported.

D. Criticality-Oriented Pattern Set Generation

In order to get a higher economic gain exploiting the non-uniform failure distribution of the device, a novel pattern set considering also the faults' criticality has been generated. Patterns are generated to target both Stuck-at and Transition delay faults. These are the fault models for the considered technology that represent the population's defectivity more. Additional reasoning would be needed for not-modeled faults.

A comparison between the pattern set currently used in production and the proposed criticality-oriented one can be now provided. To follow the ranking of the faults' criticality for the pattern generation, the *criticality-oriented ATPG* algorithm described in Algorithm 2 has been used. As many patterns have been generated as those used in production.

Fig. 16 shows the coverage obtained with the proposed criticality-oriented methodology compared to the pattern set currently used in production. At around 50% of applied patterns, the original pattern set overcomes the criticality-oriented one in terms of coverage. Such behavior is expected because the Algorithm 2 strategy requires more ATPG efforts. Thus, it is reasonable that the trend will rise slower after the most critical faults have been targeted. To reach the highest possible fault coverage with the new generation, after the completion of the generation following the ranking of faults (at around 75% of the original patterns), an additional pattern generation (*abort limit*

TABLE II
SUMMARY COMPARISON AMONG TEST COST REDUCTION METHODOLOGIES

Method	Economic Gain [%]	Type	Tester Configuration	Test Escapes Handling	Data Input	Outcome	Reported Case Study
[5]	No gain	Static	Single-site	Not handled	Statistical data from production	Reordered patterns based on production	ISCAS'89 benchmarks [23]
[6]	0.8	Adaptive	Single-site	No test escapes	Information about the previous tested wafer	Reordered patterns at each tested wafer	ISCAS'89 benchmarks [23]
[10]	1.1	Static	Single-site and multi-site	Not handled	Statistical data from production	Prediction of useless patterns	IBM devices
Proposed methodology	6.1	Static	Single-site and multi-site	Covered by successive test phases	Circuit information and statistical data from production	Model to find the best point to cut patterns	Academic benchmark and $40nm$ real-world Automotive SoC

set to 1000, *compaction effort maximum*, *coverage effort high*) on the residual faults is conducted.

Nevertheless, the key point in the strategy is to shift the most critical faults to the left as much as possible.

This is even more evident when comparing the resulting test costs in Fig. 17. By having a steeper increase in fault coverage during the initial section of the curve, the cost can be reduced more significantly if the number of schemes applied is reduced even more than in the original curve.

Considering the new generated pattern set, the cost of the scan tests is optimized. Indeed, in Fig. 17, the maximum economic gain reachable using such patterns is about 6.1% per device that is almost two times the gain achievable by reducing the original pattern set, generated without any guide regarding the failure distribution.

E. Comparison With Another Test Cost Model [10]

As described in Section II, in [10] the authors discuss about the possible useless patterns used in production by industries. The study proposes a model that takes into account the number of tested devices, the production yield and a function (F_D) representing for each pattern how many tested devices have been found as failed. For completeness, the equation for calculating the number of useless patterns is reported in Equation 14, where T is the total number of patterns, N is the device population and Y is the production yield.

$$E = \sum_{k=1}^{T} (1 - F_D(k) + F_D(k-1))^{N \cdot (1-Y)} \qquad (14)$$

In Fig. 17 also the points individuated by the model proposed by [10] are reported. In particular, their proposed formula is used to estimate the number of useless patterns for both the pattern set used in production and the one generated through the criticality-oriented ATPG algorithm. To extrapolate the F_D function, production data over six months were collected, and as shown in Figs. 14 and 15, the non-uniform model accurately predict the production trend (Table I). For this reason, it is possible to also use the proposed model to approximate the F_D function for both pattern sets.

The result is that about 86.5% of patterns belonging to the pattern set currently used in production would be useful. Thus, cutting all the other patterns would bring to a gain of about 1.1%. Meanwhile, for the patterns generated by following the proposed methodology, about 26% of patterns should be actually useful, bringing to an economic gain of about 5%.

In the next subsection, a further comparative analysis with state-of-the-art test cost reduction strategies is detailed.

F. Summary Comparison With State-of-the-Art Test Cost Reduction Methodologies

State-of-the-art test cost reduction methodologies mostly rely on pattern reordering [5], [6]. Arranging patterns to have the most effective ones first impacts the final cost, but the maximum achievable economic gain is strictly limited to the failed devices. The good devices would, in any case, be tested with the complete pattern set, making the methodology strictly limited to single-site testing. Contrarily, the proposed method and the one described in [10] propose a strategy to reach economic gain on all the devices, including the good ones. Companies manufacturing products with high yields, and a few failed devices, would benefit more from using a methodology that reduces the costs for all devices, not just the failed ones.

Table II shows a summary comparison among such test cost reduction methodologies. First, it should be noted that the proposed methodology has the highest economic gain, considering using the criticality-oriented pattern set. Moreover, it is the only one adaptable in single and multi-site testing environments that handles the potential extra test escapes in the successive test phases. Real production data for the industrial case study also support the proposed study. Finally, it not only gives a model adaptable to any pattern set already in use but also provides, as a by-product, a new one that considers the device's criticality to maximize the economic gain.

G. Computational Time and Resources

Table III details the computational time and resources needed for the proposed methodology, categorized into points A and B shown in Fig. 6. The *calculation of the candidate faults* phase is the longest. It includes fault simulation for all patterns unless the lists of covered faults by every pattern are already available. The *analysis of the candidates' criticality* phase refers to the

TABLE III
COMPUTATIONAL TIME AND RESOURCES NEEDED

Circuit	Gates [Million]	Calculation of Candidate Faults		Analysis of Candidates Criticality		Weights' Computation		Cost Modeling	
		Time [d]	Threads [#]	Time [s]	RAM [GB]	Time [s]	RAM [GB]	Time [s]	RAM [GB]
Academic	1	0.5	16	85	2	$\ll 1$	2	$\ll 1$	0.25
Industrial	20	6	32	180	8	$\ll 1$	8	$\ll 1$	0.25

density analysis of the layout. The *weights' computation* phase crosses the previously computed data to find the criticality level of the candidate faults. Finally, the proposed cost model processes all this information quickly. For a more complex device with approximately 90 million gates, model creation can take up to fifteen days using 64 threads. Despite the lengthy duration, this one-time operation is reusable for other devices with the same technology.

V. CONCLUSION

This paper investigates the trade-offs between reducing test patterns during Wafer Sort, increasing packaging costs, and screening more chips during Package Tests. A comprehensive cost model is introduced. It accounts for the non-uniform distribution of defectivity, exploiting sacrificial lots to extract helpful insights. The key findings of the study are: 1) strategically reducing the number of patterns to apply along the Wafer Sort can lead to economic gains under specific conditions of yield, fault coverage, and production costs; 2) sacrificing a set of devices can be helpful to understand the failure distribution over the die population; 3) generating patterns considering the failure distribution ensures that even when cutting patterns from the tail, the remaining ones still cover the most failure-prone gates, increasing the economic gain.

The presented results on both an academic benchmark and a real-world Automotive SoC from STMicroelectronics validate the effectiveness of the proposed methodology.

One possible limitation of implementing the full proposal in productive environment lays on the criticality-oriented pattern generation, because of the required ATPG efforts. Nonetheless, the proposed cost model is capable of predicting accurately the best point to cut patterns for any pattern set already used in production, as shown with the comparison of real production data collected over six months. In conclusion, the core principles behind the comprehensive cost model are not confined to the Wafer Sort and Package Test phases. While this study focuses on these stages, the model's adaptability extends to other test phases within the overall testing flow, inspiring further exploration and application.

REFERENCES

[1] R. Srivastava et al., "Soc time to market improvement through device driver reuse: An industrial experience," in *Proc. Int. Symp. Electron. Syst. Des.*, 2012.

[2] M. Rehani et al., "Ate data collection-a comprehensive requirements proposal to maximize roi of test," in *Proc. Int. Conf. Test*, 2004.

[3] V. Tancorre et al., "eFlash MCUs multi-temperature coverage maximization and test cost optimization," in *Proc. Int. Test Conf.,* 2015.

[4] C. He et al., "Wafer level stress: Enabling zero defect quality for automotive microcontrollers without package burn-in," *Proc. IEEE Int. Test Conf.*, 2020, pp. 1–100.

[5] T. Song et al., "Pattern reorder for test cost reduction through improved svmrank algorithm," *IEEE Access*, vol. 8, pp. 147965–147972, 2020.

[6] G.-Y. Lin et al., "A test-application-count based learning technique for test time reduction," in *Proc. VLSI Des., Automat. Test(VLSI-DAT)*, 2015, pp. 1–4.

[7] S. Goel et al., "On-chip test infrastructure design for optimal multi-site testing of system chips," in *Proc. Des., Automat. Test Europe*, vol. 1, 2005, pp. 44–49.

[8] S. Bahukudumbi, and K. Chakrabarty, "Defect-oriented and time-constrained wafer-level test-length selection for core-based digital SoCs," in *Proc. IEEE Int. Test Conf.*, 2006, pp. 1–10.

[9] M. Grady, B. Pepper, J. Patch, M. Degregorio, and P. Nigh, "Adaptive testing - cost reduction through test pattern sampling," in *Proc. IEEE Int. Test Conf.*, 2013, pp. 1–8.

[10] F.-F. Ferhani, N. R. Saxena, E. J. McCluskey, and P. Nigh, "How many test patterns are useless?" in *Proc. 26th IEEE VLSI Test Symp. (VTS)*, 2008, pp. 23–28.

[11] T. Williams, and N. C. Brown, "Defect level as a function of fault coverage," *IEEE Trans. Comput.*, vol. C-30, no. 12, pp. 987–988, Dec. 1981.

[12] F. Corsi et al., "Defect level as a function of fault coverage and yield," in *Proc. ETC 93 3rd Eur. Test Conf.*, 1993, pp. 507–508.

[13] I. Polian et al., "Exploring the mysteries of system-level test," in *Proc. IEEE Asian Test Symp.*, 2020.

[14] "OpenRISC," 2024, [Online]. Available: https://openrisc.io/

[15] A. Benso et al., "ATPG for dynamic burn-in test in full-scan circuits," in *Proc. Asian Test Symp.*, 2006.

[16] P. Wang, A. M. Gharehbaghi, and M. Fujita, "An automatic test pattern generation method for multiple stuck-at faults by incrementally extending the test patterns," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 39, no. 10, pp. 2990–2999, Oct. 2020.

[17] M. Fujita et al., "Incremental ATPG methods for multiple faults under multiple fault models," in *Proc. 16th Int. Symp. Qual. Electron. Des.*, 2015, pp. 177–180.

[18] P. Maxwell, F. Hapke, M. Ryynänen, and P. Weseloh, "Bridge over troubled waters: Critical area based pattern generation," in *Proc. IEEE Eur. Test Symp.*, 2017, pp. 1–6.

[19] F. Hapke and P. Maxwell, "Total critical area based testing," in *Proc. IEEE Int. Test Conf. (ITC)*, 2018, pp. 1–10.

[20] F. Hapke and P. Maxwell, "Defect-oriented test: Effectiveness in high volume manufacturing," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 40, no. 3, pp. 584–597, Mar. 2021.

[21] G. Iaria et al., "A novel pattern selection algorithm to reduce the test cost of large automotive systems-on-chip," in *Proc. IEEE 23rd Latin Amer. Test Symp. (LATS)*, 2022, pp. 1–6.

[22] E. Marinissen et al., "A set of benchmarks for modular testing of socs," in *Proc.. Int. Test Conf.*, 2002, pp. 519–528.

[23] F. Brglez et al., "Combinational profiles of sequential benchmark circuits," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, vol. 3, 1989, pp. 1929–1934.

[24] P. Bernardi, L. Cardone, G. Iaria, D. Appello, G. Garozzo, and V. Tancorre, "About the correlation between logical identified faulty gates and their layout characteristics," in *Proc. 29th IEEE Int. Symp. -Line Testing Robust Syst. Des.*, 2023, pp. 1–7.

[25] C. P. Robert et al., *Monte Carlo Statistical Methods*, 2nd ed., ser. Springer Texts in Statistics. New York, NY, USA: Springer 2004.

[26] W. Ruggeri et al., "Innovative methods for burn-in related stress metrics computation," in *Proc. Int. Conf. Des. Technol. Integr. Syst. in Nanoscale Era*, 2021.

Giusy Iaria (Member, IEEE) is currently working toward the Ph.D. degree in computer and control engineering with the Polytechnic of Turin, Italy. She is part of the CAD & Reliability Research Group with a strong focus on embedded electronics testing.

Paolo Bernardi (Senior Member, IEEE) received the M.S. and Ph.D. degrees in computer science in 2002 and 2006, respectively. He is an Associate Professor with the Politecnico di Torino University, where he works in the Electronic CAD and Reliability Research Group. His research interests include system-on-chip tests and reliability. He is the Program Chair of the IEEE International Test Conference 2025 and he has been the General Chair of the IEEE European Test Symposium 2023.

Claudia Bertani graduated in electronic engineering with the Politecnico di Milano. She is a Product & Test Engineer Manager with the STMicroelectronics, with 20+ years of expertise in the semiconductor industry. She is currently managing the team in charge of System-Level Test introduction on Automotive ADAS SOC products.

Lorenzo Cardone (Graduate Student Member, IEEE) received the bachelor's degree in computer science in 2021 from the Politecnico di Torino, where he is currently working toward the Ph.D. degree. His main research topics are software parallelization and optimization, and he has recently started working in the field of hardware testing.

Giuseppe Garozzo was born in Catania, Italy, in 1972. He received the M.S. and Ph.D. degrees in physics from the University of Catania, in 1996 and 2001, respectively. Since 1998, he has been employed at ST-Microelectronics and works on data analysis and management. He is the Author of some papers on plasma process simulation and statistical process control.

Vincenzo Tancorre received the M.S. degree in electronics engineering from the Politecnico di Bari. He is a Yield Enhancement Engineer with the STMicroelectronics with 20+ years of expertice in large SoC test for Automotive applications. His research interests include test-related process monitoring using diagnostic solutions for memories and unstructured logic based on DfT methodologies.

Open Access funding provided by 'Politecnico di Torino' within the CRUI CARE Agreement