Innovative methods for Burn-In related Stress Metrics Computation

Innovative methods for Burn-In related Stress Metrics Computation

W. Ruggeri, P. Bernardi, S. Littardi, M. Sonza Reorda
Dipartimento di Automatica e Informatica – Politecnico di Torino, Turin, Italy

D. Appello, C. Bertani, G. Pollaccia, V. Tancorre, R. Ugioli
STMicroelectronics, Agrate Brianza, Italy

Abstract: *Burn-In equipment provide both external and internal stress to the device under test. External stress, such as thermal stress, is provided by a climatic chamber or by socket-level local temperature forcing tools, and aims at aging the circuit material, while internal stress, such as electrical stress, consists in driving the circuit nodes to produce a high internal activity; in conjunction with several voltage conditions, such a stress can possibly lead to a break in imperfect devices. To support internal stress, Burn-In test equipment is usually characterized by large memory capabilities required to store precomputed patterns that are then sequenced to the circuit inputs.*

Because of the increasing complexity and density of the new generations of SoCs, evaluating the effectiveness of the patterns applied to a Device under Test (DUT) through a simulation phase requires long periods of time. Moreover, topology-related considerations are becoming more and more important in modern high-density designs, so a way to include these information into the evaluation has to be devised.

In this paper we show a feasible solution to this problem: the idea is to load in the DUT a pattern not by shifting inside of it a bit at a time but loading the entire pattern at once inside of it; this kind of procedure allows for conservative stress measures, thus it fits for stress analysis purposes. Moreover, a method to take the topology of the DUT into account when calculating the activity metrics is proposed, so to obtain stress metrics which are able to better represent the activity a circuit is subject to.

An automotive chip accounting for about 20 millions of gates is considered as a case of study. Resorting to it we show both the feasibility and the effectiveness of the proposed methodology.

1. Introduction

The purpose of the Burn-In (BI) process is to activate infant mortality (early life latent defects) that naturally affects populations of electronic devices. Burn-In is a manufacturing test phase used for many mission-critical modules, such as automotive microcontrollers and SoCs, which are the objective of the present research. In this field, BI steps are very useful for facing the constraints coming from safety standards such as IEC 61805 [1] and ISO 26262 [2].

A BI tester applies two types of stress. The former is the *external stress*, which is mainly based on higher temperature and higher voltage than in user mode [3][4][5]. This kind of stress is commonly introduced by means of a climatic chamber, which warms the chips up to their specification limits, and by tunable voltage regulators mounted on the cold part of the test equipment to introduce voltage margins. This type of stress is directly related to Arrhenius's law about material aging. The latter is the i*nternal stress*, which is produced by activating during the BI phase the different operational modes of the device under test (DUT) [6][7]. The main idea of BI is to combine external and internal stress in order to accelerate the activation of extrinsic defects under the bathtub curve hypothesis. Internal electrical stress can be driven through the DUT by means of the JTAG interface. The interface allows to write suitable values into the Test Data Registers that configure the Boundary-Scan.

This paper provides two main contributions: first, a "deductive" approach to speed-up internal stress simulation by means of parallel pattern loading (as opposed to serial pattern loading, which is a sort of "exhaustive" approach) is illustrated and theoretically motivated. Secondly, a topology-based metric refinement method which accounts for the circuit density is proposed.

Experiments on the proposed approach are reported which have been performed on a real world automotive processor. The device configuration utilized during the experiments is the so-called *Burn-In Test Mode* which uses a single scan chain composed of almost 700K Flip-Flops that runs through the entire system and uses the boundary scan TDI pin to shift inside the DUT the desired test pattern. To verify the validity of our approach, we performed the following experimental activities:

- We analyzed the required simulation times for different amounts of patterns using the deductive approach and compared the results to the estimated simulation times of the exhaustive approach.
- We calculated the stress metrics for different amounts of patterns both in a "naïve" way and in a *density-aware* way, then comparing the obtained results.

The obtained results show that our deductive approach allows to considerably reduce simulation times, while our density-aware metrics refinement allows to obtain insights on how the various parts of the design-under-test are stimulated by the test patterns.

The paper is structured as following: section 2 discusses the practical and theoretical background of the analyzed problems; section 3 details the proposed approach to activity simulation and stress metrics refinement; section 4 discusses results of the experiments performed on the selected case study; section 5 concludes the paper by summarizing the contributions discussed in the previous sections.

2. Background

Fig. 1 illustrates the workflow required to compute stress metrics starting from the simulation of the circuit: the circuit is simulated for a set of test patterns (which can be scan vectors, for example, or functional programs) and a simulation dump is produced, and the simulation dump is analyzed by ATPGs or by fault simulation engines (if they are suitably instructed) or by ad-hoc tools[11], so to produce a list of test/stress metrics showing how the design-under-test is stimulated.

The stimulation provided by the BI tester to the chip is typically based on the usage of scan chain(s). BI patterns add logical stress to the electrical burden of the chip in the

form of higher supply voltages and higher temperature inside the climatic chamber the chips are put in.

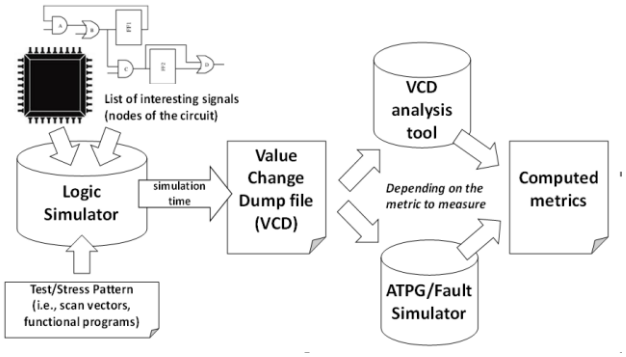

Fig. 1. Evaluation flows of metrics in case either ATPG/Fault simulator can be used or ad-hoc tools are available.

Such an extra stress aims at exacerbating latent defects of various nature by moving the circuit in specific "highly-stressing" logic conditions, possibly reflecting in a physical additional fatigue for the DUT.
This process addresses the problem of infant mortality failure, and its major purpose is to make weak devices break, thus allowing the manufacturer to screen them out before they are shipped to the market.

2.1. Stress metrics

The analysis of a Value Change Dump (VCD) file can be performed by ATPG engines (essentially resorting to their fault simulation features) to evaluate the coverage of several test and stress metrics. The toggle activity and several other fault coverage metrics (such as stuck-at, transition delay, bridges, etc.) are usually returned [8][9][10]. Unfortunately, a limitation of modern state-of-the-art ATPG engines is related to the limited amount of models and metrics they can compute. Moreover, reports collected by the ATPG sometimes provide limited statistics and information details. For example, the activity of a circuit is not always analysed over time. Therefore, there are cases in which ATPGs cannot help too much and we need to use more specific VCD-based approaches. In our particular field of application, ATPG engines typically lag behind in terms of ability to measure stress metrics related to the BI test process. Hence, suitable ad hoc tools may be introduced to post-process VCD files and compute stress metrics [11].

In details, the BI stress metrics that are evaluated in a production flow are based on the following:

- *Single point stress metrics*: An example of this metric is the very common measure of the toggle activity. This simply indicates if a circuit node holds both logic values '0' and '1' during the simulation. If this happens, the node is covered. Single point stress metrics may be enriched by adding some of the following features:
  - Extended statistics: we compute the number of times a node toggles during the simulation. This is a feature that is rarely included in ATPG engines.
  - Timing related measurement: we consider the logic behaviour of the circuit along time, i.e., we compute the average toggling frequency. This is sometimes important to ensure a similar level of stress for all nodes of the circuits.
- *Multiple points stress metrics*, which are based on the evaluation of the logic values of adjacent nodes. This kind of metrics stems from the following consideration: if the layout of the circuit is known, it is possible to extract a list of all the "neighbours" of each node, i.e., the nodes which are placed within a specified distance from that node.

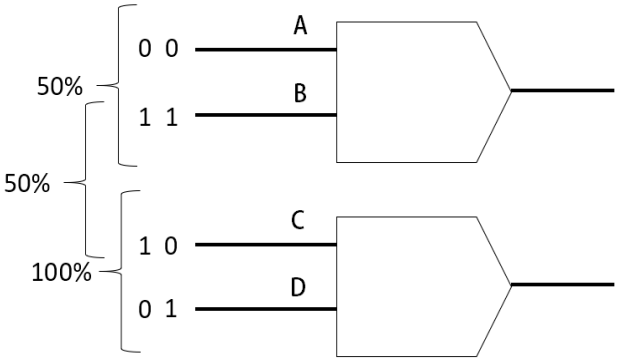

Fig. 2: A possible multiple points stress metric explanation.

This way a new "node list" can be generated containing all the possible couples of neighbours in the DUT, where the stress model is defined as follows: a couple is said to be 50% covered when a configuration of opposite values is observed on the nodes, while it is said to be 100% covered when both configurations of opposite values are recorded; this concept is illustrated in Fig. 2. This measure has similarities with the bridging fault concept, but ATPG cannot provide extended statistics and timing related measurement.

2.2. SoC topology and gate density over the chip surface

When dealing with a complex System-on-Chip, it is fundamental to understand first its topology, i.e., how it is physically organized, in order to gather useful insights that can help in understanding the meaning of the computed stress metrics and devising tests to properly cover all the parts of the chip.

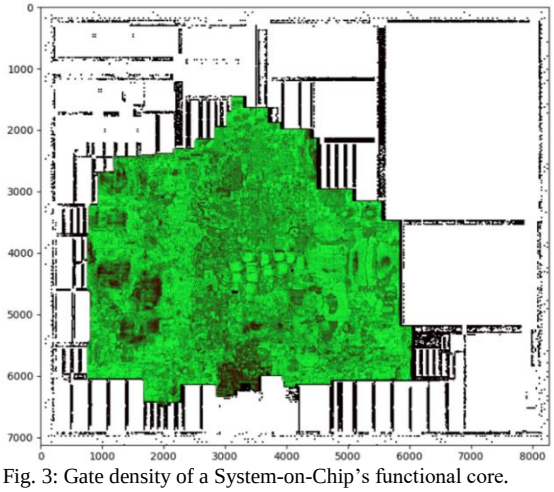

Fig. 3: Gate density of a System-on-Chip's functional core.

A generic layout of a System-on-Chip can be seen in Fig. 3, where an important concept can be highlighted: typically, a System-on-Chip does not show a uniform distribution of gates on its surface. Its main functional core includes denser areas which form a "sea of gates" and sparser parts in which just a few gates are placed. Moreover, there are large parts of the chip (the white ones

in the figure) which are dedicated to components such as memories and to the interconnection between those components and the main functional core of the System-on-Chip.

The difference in density of the various parts of a System-on-Chip is further highlighted in Fig. 3, where the different gate densities of the various parts of the sea of gates are analysed: in the figure, a brighter shade of green describes parts with a higher gate density, while a darker shade of green indicates zones in which less gates have been placed.

3. Proposed approach

In the following, the main points of the proposed approach will be presented:

- A method to evaluate a DUT activity during simulation without resorting to full scan simulation will be presented; more precisely, a method which relies on the parallel load of the whole scan chain is illustrated.
- A way to refine the stress metrics used to evaluate the circuit activity will be discussed and applied both to multi-point and single-point stress metrics. The refinement is based on the knowledge of the specific design-under-test physical structure and is defined in such a way that it can be applied to every possible stress metric beyond the ones considered in this work.

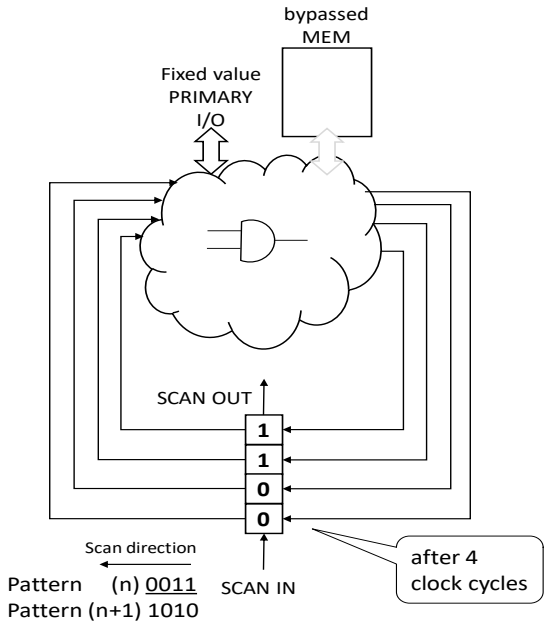

Fig. 4: Shift-in of the 1st pattern, taking 4 clock cycles.

3.1. Activity evaluation methods

To evaluate the effectiveness of a scan pattern in terms of stressing capabilities, the most straightforward solution would be to exhaustively simulate the whole shifting phase of the test vectors inside the scan chain.

This method is able to replicate exactly what happens at the hardware level, and it could be defined as a "exhaustive" approach because it measures every single activity produced by the scan shift till the last shift, when a final state is reached, the scan chain value is applied to the combinational logic, and the response is captured by the scan chain.

Once the scan chain is uploaded, every single shift applied through the TDI of the boundary scan interface causes the whole system to evolve to a new state, as shown in Fig. 4 and 5 for a small circuit including 4 FFs and bypassed memories.

Given the typical size of the devices under test, which can count millions of flip-flops in a scan chain, and up to hundreds of millions of nodes, the simulation of each clock cycle during the shift phase can take a huge amount of memory, while the simulation time increases drastically. In conclusion, the exhaustive approach, computing how many nodes toggle at each clock cycle during the whole shift phase, is practically unfeasible.

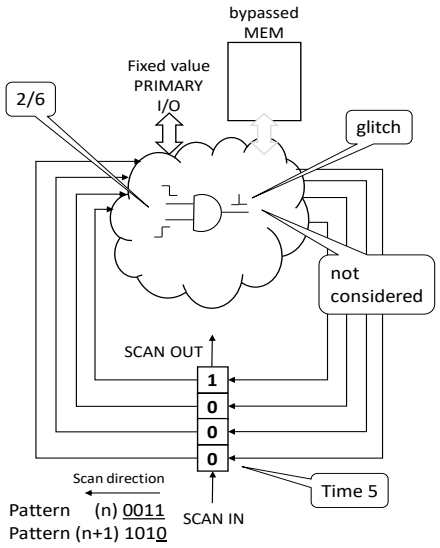

Fig. 5: Transitions created by shifting the first bit of the 2nd pattern: only 2 out of 6 transitions are covered.

In order to overcome these problems and to find a compromise between computational effort and result accuracy for simulation, a *deductive* approach is adopted, which is well-known in the ATPG engines implementation field and is here reconsidered to solve the simulation issues. Such a parallel load approach is based on the following simplifying assumption: given a set test patterns to be applied to the device-under-test, only the final configuration of the scan chain after the entire shift of every single pattern is being considered.

Theoretically speaking, it is true that, if the application of two consecutive final configurations to the combinational part causes transitions, then the same transitions will show up during shift operations. It means that:

$$\text{transitions}(\textit{deductive}) \subseteq \text{transitions}(\textit{exhaustive})$$

This consideration is the reason why the adopted approach is called "deductive": given that the recorded transitions constitute a subset of the transitions that take place when applying the exhaustive approach, a lower boundary on the possible transition coverage is actually deduced.

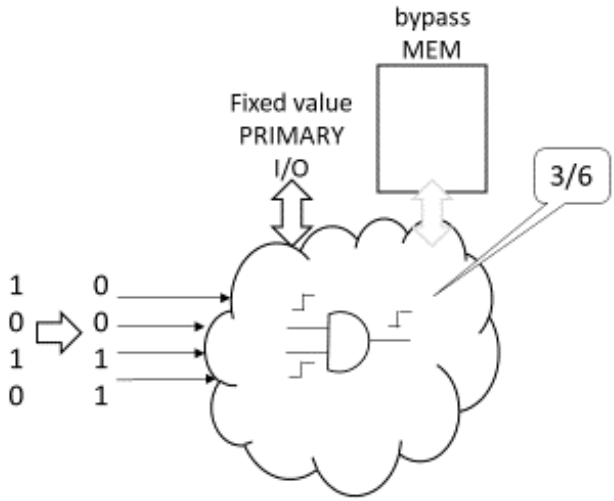

Fig. 6: Transitions created by the deductive application of two patterns: 3 out of 6 possible transitions are covered.

When simulating two consecutive final configurations only (as in the deductive approach), if we observe a generic node to toggle we can demonstrate that this node is guaranteed to toggle also considering the exhaustive approach. Hence, the node is guaranteed to toggle in practice. On the other hand, if the node does not toggle according to the deductive approach, we cannot guarantee that it does not toggle in practice. As a conclusion, the results obtained by means of the deductive approach are approximated, but conservative.

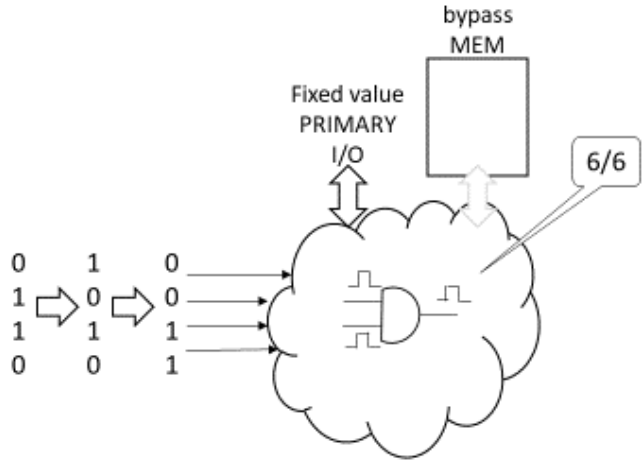

Fig. 7: Transitions created by the deductive application of three patterns: all the 6 possible transitions are covered.

Fig. 6 and 7 show graphically how the deductive approach works. Even though the deductive approach provides an approximation, the measure it provides is valuable not only for the affordable execution time: in fact, it is conservative and can guide the test engineer to create the best patterns to apply, leaving the shift phase out of consideration. This is appropriate also in terms of switching frequency, which could be high at the apply/capture time if driven by a PLL, and low (therefore less significant) during the shift phase.

3.2. Topology-related metrics refinement

Traditional stress metrics are usually "node-based", i.e., they consider the behavior of a node or of a set of nodes regardless of how the circuits they are part of is structured; they are also "unweighted", i.e. they consider each node to yield the same contribution to the metric computation. We propose to exploit the knowledge of the specific structure of a DUT, which can be fundamental in assessing how the circuit is stimulated, to enhance the considered stress metrics by taking into account the topology of the circuit.

Given a fixed inter-node distance, all the nodes of the circuits can be organized into couples of neighbors, and then a measure of the average density $\overline{D}$ of the design-under-test can be calculated by considering that each of its M nodes belongs to N_i couples:

$$\overline{D} = \frac{\sum_{i=1}^{M} N_i}{M}$$

Starting from this density estimate, one can calculate a density-weighted coverage metric for the whole chip by using the following formula:

$$C = \frac{\sum_{i=1}^{M} F(i)}{\sum_{i=1}^{M} G(i)}$$

Where the two node-related functions are defined so that the nodes belonging to the denser parts of the chip have more weight:

$$F(A) = \begin{cases} C(A) \cdot N(A) \cdot \dfrac{1}{\overline{D}} & if\ N(A) < \overline{D} \\ C(A) \cdot N(A) & if\ N(A) = \overline{D} \\ C(A) \cdot N(A) \cdot \overline{D} & if\ N(A) > \overline{D} \end{cases}$$

$$G(A) = \begin{cases} N(A) \cdot \dfrac{1}{\overline{D}} & if\ N(A) < \overline{D} \\ N(A) & if\ N(A) = \overline{D} \\ N(A) \cdot \overline{D} & if\ N(A) > \overline{D} \end{cases}$$

where C(A) is the (unweighted) coverage of node A, N(A) is the number of couples it belongs to and $\overline{D}$ is the average density of the DUT. Using these formulas, nodes belonging to the denser parts of the chip have their influence on the coverage boosted by the product between the average density factor $\overline{D}$ and the number of couples they are part of $N(A)$, while node belonging to less dense parts of the chip have their influence reduced by the ratio between the average density and the number of couples they are part of.

This kind of metric refinement can be applied both to the toggle activity metric, in which case C(A) is simply the toggle coverage of each node A, and to the multi-point metric, in which case given a node A belonging to N couples each one characterized by a coverage C_i its coverage can be defined as:

$$C(A) = \frac{\sum_{i=1}^{N} C_i}{N}$$

More generally, the refinement we propose can be applied to every possible metric: as long as a coverage measure can be defined for each node of the circuits, the proposed equations can be applied without any modifications, because they are based only on the topology of the circuit and on the requirement that a coverage metric can be provided for each of its nodes.

4. Experimental results

In the following, experimental results will be reported and analyzed, which show how the proposed approach is convenient in terms of simulation times and stress metrics analysis with respect to the traditional approaches. As a case study, an automotive microprocessor belonging to the STMicroelectronics SPC58 family has been used; the selected processor features multiple cores, many general-purpose and special-purpose modules such as timers and communication modules and a scan chain composed of almost 700k flip-flops. As such it is a good case study to analyze how the proposed approach performs in a real world setting.

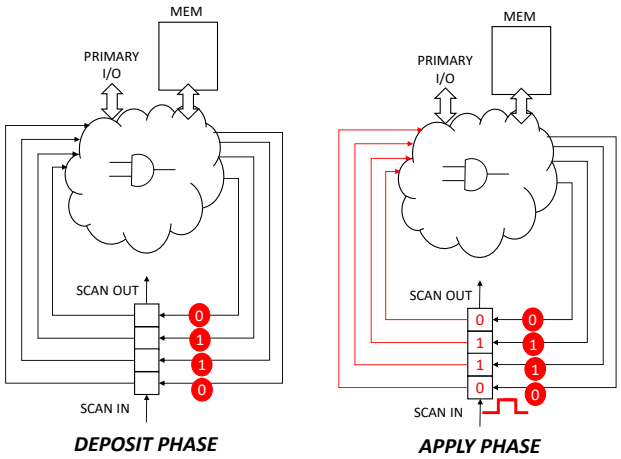

Fig. 8: Deductive simulation approach.

The deductive approach to simulation has been implemented in a two-phase logic simulation setup as shown in Fig. 8. The first phase is called *deposit phase*, and it takes place during the period when the clock signal is low, when a test vector is loaded in parallel in the scan chain by simulation.

After a period, in the *apply phase*, the scan enable is driven to the 0 logical value putting the device in a functional state and, before reaching the next rising front of the clock, the scan enable is brought back to the 1 logical value, thus moving the system back to scan mode.

The process is repeated for each test vector forming a pattern, as illustrated in Fig. 9. Once each test vector is deposited in the scan chain, the results are extracted from a VCD file generated during the simulation. This file reports for each node state how many toggles happened in the observed time.

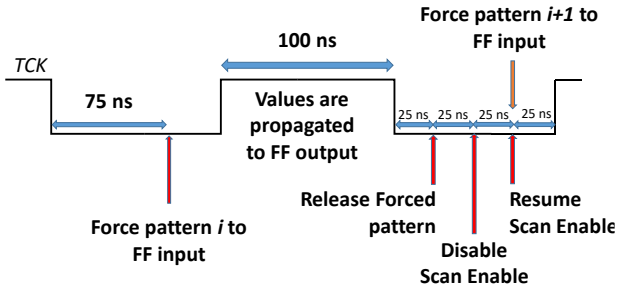

Fig. 9: Timing details about the deductive simulation approach.

The VCD file produced by a logic simulator is then analyzed via several scripts and tools, dumping all activities related to a selected set of signals; in our case, we selected all signals of the DUT and the simulation produced a file whose size may be up to 250GB depending on the amount of simulated test vectors and on the amount of activity triggered inside the DUT. In terms of CPU time, a single vector simulation and analysis takes around 12 minutes.

We have adopted the deductive approach and performed the measurement for 32 test vectors in 3 hours on a single core of a server equipped with a 64-bit 16-core processor and a 128 GB RAM memory running a Linux-based operating system. The estimation of the measurement time by adopting the exhaustive approach is around 5,376 hours (about 7 months). Of course this number is an estimation and, even if approximated, demonstrates that the precise measurement is absolutely unfeasible.

Table I: Simulation CPU times

Test vectors	Exhaustive (estimated)	Deductive
32	5,376 hours	3 hours
128	21,504 hours	12 hours
1,024	172,032 hours	96 hours

Table I provides more details about computation times, all of them including simulation and post processing of the dumped information.

In order to assess the real extent of the difference between the exhaustive approach and the deductive approach, experiments have been performed on an open-source benchmark design (the OpenRisc 1200), small enough to allow the exhaustive simulations to be actually performed. The results are shown in Fig. 10, which highlights that the difference in terms of stress coverage between the exhaustive and the deductive approach is very low, amounting to less than 0.2% when about 200 patterns are applied. These results prove the validity of the deductive approach, which as expected is able to provide precise enough results with reasonable computational effort.

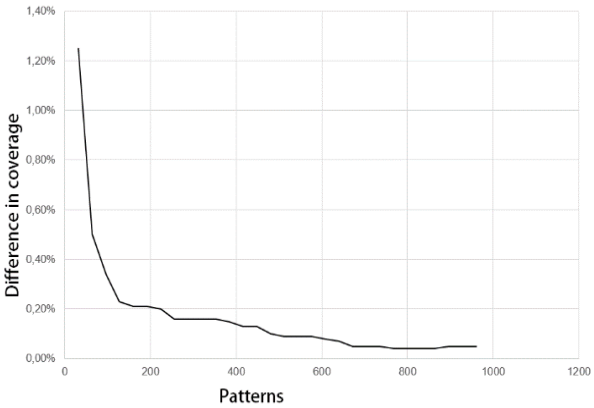

Fig. 10: Difference in coverage between the exhaustive approach and the deductive approach to simulation.

As for the evaluation of the stress metrics, experiments have been performed to show how topology awareness affects the activity metrics. The results for the toggle activity are detailed in Table II, while the results for the multiple-point metric are detailed in table III.

Table II: Toggle activity coverage

Test vectors	Unweighted coverage	Density-weighted coverage	CPU time
1	46%	44.68%	12 m
2	64%	62.98%	13 m
4	74%	73.78%	16 m
8	79%	79.51%	23 m
16	83%	83.72%	43 m
32	86%	87.79%	55 m
64	88%	89.81%	1.7 h
128	91%	93.42%	4.2 h

Table III: Multiple point static stress metric coverage

Test vectors	Covered couples	Average node coverage	Density-weighted coverage	CPU time
1	44%	45.34%	44.29%	17 m
2	57%	57.81%	57.05%	23 m
4	66%	66.29%	66.19%	26 m
8	72%	71.84%	72.39%	1 h
16	76%	75.61%	76.81%	2.5 h
32	80%	78.89%	80.85%	6.2 h
64	82%	80.54%	82.94%	19.2 h
128	85%	83.24%	86.36%	66.3 h

Fig. 11 visually represents how the weighted and unweighted activity metrics evolve with respect to the number of applied patterns: it can be seen that when just a few patterns are used the density-weighted activity is lower than the unweighted one, while when more than 8 patterns are applied the density-weighted metric tends to have higher values, which means that a much greater part of the denser areas of the DUT are being covered.

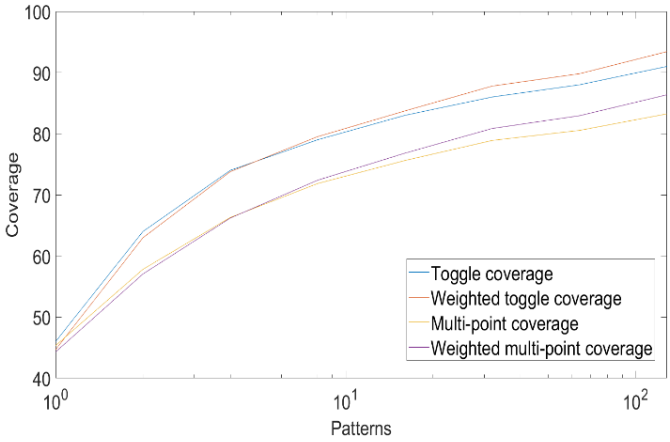

Fig. 11: Evolution of the unweighted and weighted toggle coverage.

This kind of behavior captures the way the patterns stimulate the different parts of the circuit: with just a few patterns applied, the stimulation is more "diffused" across the circuit, while when many patterns are applied the activity tends to be centered in the denser parts of the chip and as such the density-aware metric tends to increase and eventually it exceeds the values provided by the unweighted metric.

5. Conclusions

In the presented research, two main methods to improve burn-in related stress analysis have been proposed: the first one is a "deductive" approach to scan simulation which is based on the parallel loading of test vectors so to avoid the considerable time effort that would be required by a full scan simulation; the second one is a topology-based refinement which allows to improve the stress metrics ability to capture the activity of a design-under-test by means of a simple weighting algorithm which considers the density of the circuit. Experiments performed on a real world case study have shown how the proposed approach is able to considerably reduce simulation time and make stress metrics more sensible to the topological characteristics of the design-under-test.

6. References

[1] International Standard - IEC 61508 - Functional safety of electrical/electronic/programmable electronic safety-related systems, International Electrotechnical Commission, 2010.

[2] ISO 26262-[1-10], Road vehicles – Functional safety, 2011.

[3] A. Benso, A. Bosio, S. Di Carlo, G. Di Natale and P. Prinetto, "ATPG for Dynamic Burn-In Test in Full-Scan Circuits," 2006 15th Asian Test Symposium, pp. 75-82.

[4] A. Birolini, "Reliability Engineering Theory and Practice," Heidelberg: Springer, 2017

[5] M. F. Zakaria et al., "Reducing burn-in time through high-voltage stress test and Weibull statistical analysis," IEEE Design & Test of Computers, vol. 23, no. 2, pp. 88-98, March-April 2006.

[6] D. Appello et al., "A comprehensive methodology for stress procedures evaluation and comparison for Burn-In of automotive SoC," Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017, Lausanne, 2017, pp. 646-649.

[7] D. Appello et al., "An Optimized Test During Burn-In for Automotive SoC," in IEEE Design & Test, vol. 35, no. 3, pp. 46-53, June 2018, doi: 10.1109/MDAT.2018.2799807.

[8] E. Armengaud, A. Steininger and M. Horauer, "Towards a Systematic Test for Embedded Automotive Communication Systems," in IEEE Transactions on Industrial Informatics, vol. 4, no. 3, pp. 146-155, Aug. 2008, doi: 10.1109/TII.2008.2002704.

[9] M. Bakiri, C. Guyeux, J. Couchot, L. Marangio and S. Galatolo, "A Hardware and Secure Pseudorandom Generator for Constrained Devices," in IEEE Transactions on Industrial Informatics, vol. 14, no. 8, pp. 3754-3765, Aug. 2018, doi: 10.1109/TII.2018.2815985.

[10] K. Croes, D. Kocaay, I. Ciofi, J. Bömmels and Z. Tökei, "Impact of process variability on BEOL TDDB lifetime model assessment," 2015 IEEE International Reliability Physics Symposium, Monterey, CA, USA, 2015, pp. BD.5.1-BD.5.5, doi: 10.1109/IRPS.2015.7112777.

[11] P. Bernardi et al., "Accelerated Analysis of Simulation Dumps through Parallelization on Multicore Architectures", Design and Diagnostics of Electronic Circuits and Systems, 2021, doi: 10.1109/DDECS52668.2021.9417048