Next Generation Design For Testability, Debug and Reliability Using Formal Techniques

Sebastian Huhn*†

Rolf Drechsler*†

*University of Bremen, Germany
{huhn,drechsler}@informatik.uni-bremen.de

†Cyber-Physical Systems, DFKI GmbH
28359 Bremen, Germany

***Abstract*—The integration of *Design for Testability* measures is strictly required when designing complex *Integrated Circuits* (ICs) to ensure that a good testability prevails in the resulting design. By this, a high-quality manufacturing test can be performed, giving a certain level of confidence that no defects have occurred during the manufacturing process, which potentially tamper with the functional behavior's correctness. However, a high-quality test implies large test data volume and high test application time, yielding high test costs. This effect is even more amplified when testing ICs for safety-critical applications like automotive systems or avionics, enforcing a zero-defect policy. Analogously, specific structures for the *Design for Debug and Diagnosis* are introduced since similar problems exist when debugging complex systems. Finally, the *Design for Reliability* is becoming increasingly important in applications like avionics since the introduced system has typically to deal with harsh environmental conditions and, hence, the IC has to exhibit a specific level of *robustness* to withstand. This paper proposes novel contributions to, in the end, pave the way for the next generation of IC, which can be successfully and reliably integrated even in safety-critical applications. In particular, this paper combines formal techniques, such as the Boolean satisfiability problem and bounded model checking, to propose (I) a novel test access mechanism with embedded compression including an optimization-based retargeting framework, (II) a new hybrid compression architecture to address compression aborts and (II) an effective fault detection mechanism for single transient faults. The proposed measures are evaluated by considering industrial-relevant benchmark candidates, demonstrating their effectiveness and showing that state-of-the-art techniques are outperformed.**

I. Introduction

Several improvements in the *Electronic Design Automation* (EDA) flow enabled the design of highly complex *Integrated Circuits* (ICs). This complexity has been introduced to address the challenging intended application scenarios, for instance, the provided functionality and further non-functional requirements like the available computing power or the resulting power profile. Consequently, the circuit test is an essential part of this EDA flow and is becoming even more relevant since the design's overall complexity increases together with the complexity of the manufacturing process itself. The integration of *Design for Testability* (DFT) measures like [1], [2] is strictly required when designing state-of-the-art ICs to, among others, ensure that a good testability prevails in the resulting design. This testability allows performing high-quality manufacturing tests, which gives a certain level of confidence that no defects have occurred during the manufacturing process, which would tamper the functional behavior's correctness. Such a defect possibly leads to disastrous consequences in the field of safety-critical systems or, even within non-critical applications, to customer returns and a crucial loss of reputation. The steadily increasing design complexity also yields a significant increase in the *Test Data Volume* (TDV) and, hence, the *Test Application Time* (TAT) during the circuit test, which both increase the test costs. This effect is even more amplified when generating tests for advanced *System-On-A-Chip* (SoC) designs that are used in safety-critical applications like advanced driver assistance systems that strictly enforces a zero-defect policy.

Various works have been published focusing on the demands of the *Automatic Test Pattern Generation* (ATPG), *Embedded Deterministic Test* (EDT) [3] or scan chain data compression [4], [5] during wafer test. For instance, EDT achieves a very high compression ratio, which mainly scales with the share of *unspecified values* (X-values) in the incoming test data. Thus, the compression ratio of EDT strongly varies with the test data and for fully-specified data, as given in *Functional Verification* (FV) test cases, it cannot be applied at all. Furthermore, techniques like EDT require access to designated IO pins that is not given at the chip-level or in *Low-Pin Count Test* (LPCT) environments like the burn-in test.

Typically, specific structural characteristics in test patterns yield compression aborts and, consequently, these affected test patterns would be dropped from the final test set. Due to the high requirements regarding the test coverage for safety-critical systems, these so-called *Rejected Test Patterns* (RTPs) are crucial for the test and cannot be neglected. Thus, these RTPs have to be applied during the test to avoid any loss of test coverage by transferring them to the chip in a sequential and completely uncompressed fashion, i.e., bypassing the compression architecture. This *bypass operation* yields an adverse impact on the overall compression ratio and a significant test cost increase.

In other works like [6], the authors have proposed a static encoding, which has been extended by taking advantage of a static run-length encoding [4]. This run-length encoding allows for handling repeating bits within the test vector effectively. The proposed method of paper [7] uses the well-known Huffman-Encoding or even the powerful *LZ77* algorithm [5], which is based on a dynamically growing dictionary. However, significantly more hardware resources are allocated. The authors in [8] introduce a new way of encoding – the *Golomb-Code* –, which takes advantage of previous parts of the test sequence. Additionally, a powerful method of compressing multiple scan chains by a dictionary-based approach was published in [9]. In particular, [10] proposes a new compression technique, which works independently of X-values. Thus, no X-values have to be injected by the ATPG tool before transmitting the test data.

In addition to these testability criteria, further requirements have to be met, for instance, the overall number of input and output pins at the chip-level, whose number heavily affects the resulting manufacturing costs. Thus, only the chip-level pin-set is accessible in later test phases, e.g., board or in-field testing. Due to the increasing number of modules within SoC designs, ensuring the accessibility of every single module is a challenging task. To tackle this problem in those SoC designs, *Test Access Port* (TAP) controllers are used as a general access interface. Typically, a highly reduced clock frequency is applied while transferring data via this TAP, which means that the TDV or the number of test cycles highly affects the overall costs. Besides these DFT measures, specific structures in the sense of *Design for Debug and Diagnosis* (DFD) are introduced since similar problems exist when debugging complex systems.

Concurrent-JTAG (CJTAG) [11] is designed to accelerate FV using TAP structures and is already realized by industrial implementations. However, CJTAG requires highly modified devices regarding their TAP controllers such that the compliance with the standardized protocol is no longer ensured. Furthermore, two additional IO pins must be embedded in the top-level. CJTAG achieves the speed-up by parallelization necessitating structural requirements for the circuit-under-test and test sequences. Additionally, the authors in [11] propose a compression scheme for FPGA configuration bitstreams, which must be integrated in-between the TAP and the FPGA core. Due to the fact that such a bitstream file is dominated by tailing zeros characteristically, this hardware is based on a run-length encoding, which is suitable for this specialized application field only.

Analogously to manufacturing defects, a transient fault occurring during the functional operation can lead to the same disastrous consequences, particularly when being involved in a safety-critical application scenario. Such a transient fault occurs in harsh environments like a high radiation level, which potentially invalidate the functional behavior. Different measures were proposed in the literature to protect a circuit against transient faults. Prominent candidates are the *Triple Modular Redundancy* [12], [13], which introduces a large area overhead of more than 3x, and approaches like *Razor* [14], [15], which heavily influences the worst-case latency of the circuit. Another important aspect concerns the resulting overall costs of the device, which is basically a contradictory objective between low costs, a reliable design and high quality testing. These circumstances inevitably require new measures to achieve the required level of testability, debug and reliability of the resulting circuit.

This paper proposes several novel approaches to, in the end, pave the way for the next generation of ICs, which can be successfully and reliably integrated even in safety-critical applications. In particular, the paper at hand combines formal techniques – like the *Boolean Satisfiability* (SAT) [16] in combination with *Pseudo-Boolean Optimization* (PBO) and the *Bounded Model Checking* (BMC) [17] – to address the arising challenges concerning the increase in TDV, TAT and the required reliability.

This paper proposes the four main contributions as follows:

- Developing *Test <u>Vec</u>tor <u>T</u>ransmitting using en<u>h</u>anced c<u>o</u>mpression-based TAP cont<u>r</u>ollers* (VecTHOR) [18]: VecTHOR proposes a newly designed compression architecture, which combines a codeword-based compression, a dynamically configurable dictionary and a run-length encoding scheme. VecTHOR fulfills a lightweight character and is seamlessly integrated within an IEEE 1149.1 – also known as *Joint Test Action Group* (JTAG) – *Test Access Port* (TAP) controller while achieving a significant reduction of the TDV and the TAT by 50%, which directly reduces the resulting test costs. Moreover, VecTHOR can even be applied on fully-specified test data, for which most test compression techniques are not applicable at all.
- Building a retargeting framework to process existing test data off-chip once prior-to the transfer without the need for an expensive test regeneration [19], [20]. Different techniques have been implemented to provide choosable trade-offs between the resulting TDV as well as TAT and the required run-time of the retargeting process. These techniques include a fast heuristic approach and a formal optimization SAT-based method by invoking multiple objective functions. On top of that, a novel (configuration) state-aware partitioning scheme [20] has been proposed that allows to process even large industrial-scaled test data.
- Designing a hybrid embedded compression architecture, which specifically addresses the challenges for LPCT in the field of safety-critical systems enforcing a zero-defect policy [21], [22]. This hybrid approach allows reducing the resulting test time by a factor of approx. three and has been realized in close industrial cooperation with *Infineon Germany*.
- Developing a new methodology to significantly enhance the robustness of sequential circuits against transient faults while neither introducing a large hardware overhead nor measurably impacting the latency of the circuit [23]–[25]. This methodology conducts application-specific knowledge by applying SAT-based techniques and BMC to achieve this, which yields the synthesis of highly efficient *Fault Detection Mechanisms* (FDMs). More precisely, BMC is adopted to analyze the state space of an arbitrary sequential circuit to determine states, in which derived *Equivalence Properties* hold that follow a newly developed concept.

The focus of this research work was to, on the one side, take advantage of codeword-based compression techniques for significantly decreasing the TDV and TAT for fully-specified data during system-level & in-field test and LPCT. On the other hand, the reliability issue of sequential circuit against transient faults is being addressed by a novel approach to synthesize highly effective FDMs.

The main contributions of this work are summarized as follows:

- Significant reduction of the TDV & TAT for fully-specified test, debug and industrial-relevant *Functional Verification* (FV) data,
- embedded compression technique for RTPs in LPCT environments, and
- strong enhancement of the robustness of arbitrary sequential circuits against transient faults while neither introducing a large hardware overhead nor measurably impacting the circuit's latency.

All presented techniques have been discussed in detail, implemented and thoroughly validated. The techniques have been published in several formats like various conferences (including ETS, VTS, DATE, ASP-DAC, ITC-Asia, and DFTS), journals [24], [26] and an entire book [27] as well, which are stated in the references. In particular, the proposed VecTHOR architecture has been tested by considering industrial-representative candidates, which clearly demonstrated the efficacy of the proposed approaches. The comprehensive retargeting framework is publicly available at GitHub under the terms of the MIT license[1]. This retargeting framework has been further cross-compiled to an *ARMv8A Cortex-A53* microprocessor target device, which allows emulating in combination with an electrical validation using a storage oscilloscope. The hardware developments have further been prototypically synthesized to a *Xilinx XCKU040-1FBVA676* field-programmable gate array.

The remainder of this paper is structured as follows: At first, Section II gives an short introduction to relevant topics of this

[1] See repository at http://unihb.eu/VecTHOR.

work, such as the robustness assessment of sequential circuits and codeword-based compression techniques, accompanied by the introduction to formal techniques such as SAT. Section III presents the first contribution that is about the codeword-based compression architecture of VecTHOR, which is then extended by an effective optimization-based retargeting framework, as presented in Section IV. Subsequently, the hybrid compression architecture is presented in Section V and the new methodology to enhance the robustness of designs is described in Section VI. Finally, the experimental evaluation for all proposed approaches is given in Section VII and the conclusion is summarized and an outlook is drawn in Section VIII.

In summary, this work clearly demonstrates that the proposed measures can be integrated into the state-of-the-art EDA flow and are capable of solving the shortcomings of existing approaches in, but not limited to, the automotive domain.

II. Background

Within the last decade, many different test compression techniques have been proposed in the literature, which all aim to reduce the test data volume and the test application time. This volume scales directly with the required memory resources on the automatic test equipment, which is one strictly limited resource, in particular when considering test cost reduction techniques like multi-site testing [28]. Besides this, different methodologies have been proposed to enhance the reliability of sequential circuits against transient faults, forming an urgent need when targeting safety-critical applications.

A. Sequential Circuits

A sequential circuit Φ is given as a commonly known gate level representation that consists of *Primary Inputs* (PIs), *Primary Outputs* (POs), combinational gates G, and *sequential elements* (SE) such as FFs, i.e., $\Phi = (\mathsf{IN}, \mathsf{OUT}, \mathsf{G}, \mathsf{SE})$. The sequential elements are assumed to be synchronous to (at least) one clock domain.[2] The FFs of a given sequential circuit can be grouped by a *hierarchical levelizing* procedure. Two FFs FF_i and FF_j are contained in the same group if the number of FFs in both fan-in cones are the same on the shortest path towards the PIs.

Alternatively, a sequential circuit can also be represented by a *Finite State Machine* (FSM). An FSM is defined by a tuple $M = (I, S, T)$, where I describes the set of initial states, S represents the state space of the circuit, and T defines the *transition relation*. A *transition relation* $T(s, s')$ evaluates to true, if there is at least one transition from state s to state s'. The *set of reachable states* $S^* \subseteq S$ contains those states that are reachable from an initial state in an arbitrary number of steps.

B. Transient Faults

The shrinking feature size leads to an increased vulnerability of circuits against *Single Transient Faults*, which are typically caused by *Single Event Upsets*, e.g., electrical noise, particle strikes, or other environmental effects [29], [30]. Typically, the influence of a transient fault occurring at a FF is modeled as an unintended toggled output value. This influence can possibly cause an invalid and unintended behavior of the circuit Φ for a short period of time. Based on this vulnerability, a circuit Φ is called robust if no fault exists such that the input/output behavior is affected. In order to increase the robustness of a circuit Φ, an FDM can be applied that handles cases in which a single transient fault occurs at a FF, e.g., to realize precautions.

C. Assessing Robustness

To consider the vulnerability of sequential circuits against transient faults, a metric for robustness has been introduced, which measures the fault tolerance (i.e., the robustness) with respect to a fault model [31], [32]. More precisely:

Definition 1. *Let $\Phi = (\mathsf{IN}, \mathsf{OUT}, \mathsf{G}, \mathsf{SE})$ be a sequential circuit. A FF is considered to be non-robust, if there is at least one reachable state and one transient fault such that the output behavior of Φ is tampered. Let N be the* set of non-robust FFs *with $N \subseteq \mathsf{SE}$. Then, the robustness of Φ can be determined by $\mathcal{R} = 1 - \frac{|N|}{|\mathsf{SE}|}$ [33].*

In order to determine the robustness of a given sequential circuit, the non-robust FFs $\mathbb{N}$ can be computed by either formal methods [34] or simulation-based techniques [35]–[38].

D. Test Access Mechanisms

At least one communication channel, for instance, between a SoC containing several sub-modules and the test equipment, has to be available. Since strong limitations regarding the number of IO pins exist, it is not possible to route every IO pin of each sub-module to the top-level entity. For that reason, a centralized master TAP is integrated into the top-level: Such a TAP provides an access port and is managed by the TAP controller, which implements a specific interface protocol. One very common protocol is implemented within a JTAG controller. This interface consists of only five pins, which have to be accessible from outside. Every additional pin would increase production costs and the necessary design effort. In the case of hierarchical SoC designs, these capabilities can be used to access specific sub-modules.

E. Formal Techniques

The *Boolean Satisfiability* (SAT) problem asks the question whether a satisfying solution for a given Boolean function exists. These functions can be represented by using a *Conjunctive Normal Form* (CNF). A CNF Φ is a conjunction of clauses, whereby, such a clause ω is a disjunction of literals and a literal represents a Boolean variable ν in its positive x or negative form $\bar{x}$. This Boolean function $\Phi : \{0,1\}^n \rightarrow \{0,1\}$ is classified as *satisfiable* (sat) if an assignment of all variables exists such that $\Phi = 1$ holds. Otherwise, it is classified as *unsatisfiable* (unsat) [39]. In fact, such a SAT problem can be used to model several (research) questions. Generally, solving these functions is a hard computational task and, hence, a lot of research work has been spent on developing powerful solving algorithms (SAT solvers) to address this challenging problem.

Example 1. *Let $\Phi = (x_1 + \bar{x}_2 + x_3) \cdot (\bar{x}_1 + x_2) \cdot (x_2 + \bar{x}_3)$. Consequently, $x_1 = 1$, $x_2 = 1$ and $x_3 = 0$ is a satisfying variable assignment.*

The *Pseudo-Boolean* (PB) SAT problem allows for an integration of weights. The PB-SAT instance $\Phi : \{0,1\}^n \rightarrow \{0,1\}$ consists of conjugated constraints $\sum_{i=1}^{n-1} c_i \cdot \widehat{x}_i \geq c_n$ using $c_1, \ldots, c_n \in \mathbb{Z}$ as weights and $\widehat{x}_i$ as positive or negative literals. Additionally, the PBO problem extends the PB-SAT problem such that an objective function $\mathcal{F}$ can be integrated to assess the determined solution's quality. By this, it is possible not only to give an arbitrary solution as regular SAT solvers do but to determine the optimal solution with respect to $\mathcal{F}$.

[2] In order to ease the following descriptions, we will assume a single clock domain. However, the proposed methodology can be extended to further clocks domains as well.

The (PB-)SAT instance, i.e., the Boolean formula in CNF or the PB constraints, respectively, is extended with an objective function $\mathcal{F}$.[3] Typically, the objective function $\mathcal{F}$ is given as a linear sum:

$$\mathcal{F}(x_1, \dots, x_k) = \sum_{i=1}^{k} m_i \cdot \widehat{x}_i \text{ with } m_1, \dots, m_k \in \mathbb{Z}$$

Basically, the result of $\mathcal{F}$ is the arithmetic sum of all constants m_i associated with a literal $\widehat{x}_i$, which evaluates to true under a given assignment. Usually, a PBO solver utilizes the minimization as a solving target, i.e., it returns the solution which minimizes $\mathcal{F}$.

Example 2. *Let* $\Phi = (3x_1 + 4\bar{x}_2 + x_3 \geq 3) \wedge (3\bar{x}_1 + 4x_2 \geq 2) \wedge (4x_2 + \bar{x}_3 \geq 4)$ *and* $\mathcal{F} = 1x_1 + 1x_2 + 1x_3$*. In this case, the solution* $x_1 = 1,\ x_2 = 1$ *and* $x_3 = 0$ *satisfies the given PB-SAT instance and, at the same time, minimizes the outcome of the objective function* $\mathcal{F}$ *(*$\mathcal{F} = 2$*). In contrast, the solution* $x_1 = 1,\ x_2 = 1$ *and* $x_3 = 1$ *also satisfies the instance, but results in* $\mathcal{F} = 3$*, which is higher than the previous solution.*

Dedicated solving algorithms exist for these kinds of problems, many of these algorithms use SAT solving techniques internally while modern PBO solvers also support multiple objective functions $\mathcal{F}_1, \mathcal{F}_2, \dots, \mathcal{F}_n$. Here, priorities are used. First, the objective function is used as $\mathcal{F}_1$ and, subsequently, the solution is improved concerning $\mathcal{F}_2, \dots, \mathcal{F}_n$. However, it is impossible to decrease the objective function's result with a higher priority. PBO-based or similar optimization-based procedures have already been successfully applied in the testing domain, e.g., in [40], [41].

III. TAP-Controller with Embedded Compression

This section describes the first contribution of this work, namely VecTHOR. VecTHOR is meant to be seamlessly integrated into a standardized IEEE 1149 TAP-controller. For realizing VecTHOR, a mechanism is required that allows activating the new compression techniques while being fully compliant with the standardized protocol. Furthermore, the core, a codeword-based *Dynamic Decompressing Unit* (DDU) and an instrument to configure the DDU – based on the test data to be processed – have to be developed. Consequently, the following extensions and components have to be implemented:

1) Two further JTAG instructions are integrated, which activate the data compression by *compr_data* and the configuration preloading by *compr_preload*. Therefore, the FSM of the underlying TAP controller has to be extended. Additionally, a further compression technique is developed in this work: *μ-compr*, which allows taking advantage of the previously sent test data in terms of run-length encoding.
2) A suitable decompressor unit is designed that allows substituting between compressed and decompressed bit strings, and connecting this unit with a data sink such as a test data register.
3) A retargeting framework is developed, allowing to process an incoming test vector automatically: At first, a suitable configuration for the DDU is derived out of the test vector. Subsequently, the incoming test data are processed in such a way that valuable parts of the original data are replaced considering the current DDU configuration. A highly effective retargeting framework is introduced in Section IV, following the works [19], [20].

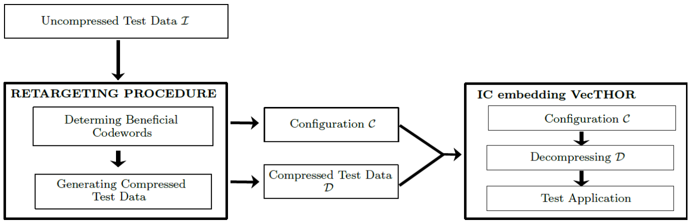
Uncompressed Test Data $\mathcal{I}$
RETARGETING PROCEDURE
Determing Beneficial Codewords
Generating Compressed Test Data
Configuration $\mathcal{C}$
Compressed Test Data $\mathcal{D}$
IC embedding VecTHOR
Configuration $\mathcal{C}$
Decompressing $\mathcal{D}$
Test Application

Figure 1: Overall compression flow of VecTHOR

Note that the above-mentioned *μ-compr* extension consists of a partial run-length encoding of complete codewords, which improves the compression ratio even more when processing homogeneous test data fractions.

A. Extension of the Finite State Machine

One very valuable property of this compression technique is that all changes performed to the interface are completely transparent toward legacy compatibility. Only two additional opcodes `0110` and `0100` are reserved, which represent the new JTAG instructions *compr_data* and *compr_preload*. Here, *compr_preload* allows to load the determined configuration to the DDU. If neither *compr_data* nor *compr_preload* instruction is loaded at all, the FSM is traversed normally and the processed data is not affected by any decompression. For the case that the instruction *compr_data* or *compr_preload* is selected, a test reset pulse trigger or even unloading this instruction enables the normal operation mode. A detailed description of the modified FSM is provided in [18].

B. Dynamic Decompressing Unit

Besides the required FSM extension, another important aspect of the proposed compression architecture is the decompressor, which realizes a mapping function $\Psi(CDW^c) \rightarrow UDW^u$ between one successfully received *Compressed Data Word* (CDW) with length c with $c \leq$ *Chunk Size* (CS) and a specific *Uncompressed Data Word* (UDW) with length u with $u \in \{1,\ 4,\ 8\}$. Additionally, the function $B(CDW^c) \rightarrow \beta$ maps every CDW to a scalar value β, which represents a metric calculated by $u - c$ and determines how valuable it would be to use this replacement - so called benefit of this replacement. Additionally, a mechanism has to be established, which ensures the completeness of the mapping function Ψ. This means that single bits, which are not directly coverable by replacements, have to be handled individually.

Consequently, the decompressor unit receives the CDW after the FSM reaches the state *compr_exit*. Reconsinder that the maximal length of the CDW is determined by the CS. For instance, a CS value of 3 offers a good trade-off between the number of possible encodings and the hardware overhead. For that reason, CS is assumed to be 3 in this work. In general, the underlying technique can be scaled with higher CS easily. For every single increment of CS, overall 2^{CS} additional ones are available. Using $CS = 3$ allows the following $\sum_{i=0}^{3} 2^i = 15$ possible CDW:

- $\varnothing$, '0', '1', '00', '01', '10', '11'
- '000', '001', '010', '011', '100', '101', '110', '111'

VecTHOR allows a so-called *Single Bit Injection* (SBI) for covering a '0' or '1' at a specific position in the test vector, which could not be covered otherwise. Consequently, the SBI

[3] Since a Boolean formula in CNF can be easily transformed into PB constraints and modern PBO solvers typically accept CNFs as input, we use the notion of CNF in this paper if possible.

enables the completeness of the compression technique, i.e., this ensures that every sequence can be successfully processed and, hence, ensures the general applicability on arbitrary test data. In the following Section IV, an entire retargeting framework is described that completes the decompressor-based TAP-controller.

IV. Optimization SAT-based Retargeting

TAP controllers with embedded compression offer a powerful mechanism for TDV and TAT reduction. However, the problem of codeword selection for a test data stream, particularly, for high-entropy data, is yet to be solved satisfactorily. This section describes the new proposed technique that addresses this shortcoming. A retargeting procedure using formal optimization-based techniques to determine an optimal configuration as well as the best sequence of replacements is presented.

The main idea is to formulate the retargeting problem as a formal optimization problem. The regular uncompressed test data $\mathcal{I}$ are given as input. An optimization problem consisting of SAT and PB constraints and an optimization function are formulated. Subsequently, a PBO solver is called to solve the problem instance. The result is a satisfying model, whose data are extracted and directly used to determine an optimal configuration $\mathcal{C}$ as well as the targeted compressed test data $\mathcal{D}$. Since the TAP controller is dynamically configurable, the calculated configuration $\mathcal{C}$ can be loaded into the controller to decompress the compressed test data $\mathcal{D}$ on-chip, which restores the original uncompressed test data $\mathcal{I}$. This technique allows a TDV reduction of the data, which have to be serially transferred into the TAP controller. This procedure can also be used for partitioned data streams due to the dynamic configuration.

The hardware constraints are assumed as follows:

1) A circuit design that embeds a codeword-based TAP controller, which includes a DDU with a dictionary consisting of $n = \sum_{i=2}^{CS} 2^i$ dynamically configurable entries.
2) Three dictionary entries are statically included, i.e., $\varnothing$, ‘0‘, ‘1‘. These encode the RLE capability and both codewords with a length of 1 for SBI modeling. Therefore, only $n-3$ codeword entries are dynamically configured by the proposed retargeting procedure.
3) Let $u = \{1, 4, 8\}$ be the UDW length supported by the *Test Data Register* (TDR) interface. Then, $2^1 + 2^4 + 2^8$ possible Boolean permutations exist. Each permutation is a candidate for being included as a codeword in the DDU.
4) The incoming test data sequence $\mathcal{I}$ contains only fully-specified values, i.e., ‘0‘ and ‘1‘.

Based on these basic conditions, a formal PBO model has to be built, which allows the automatic computation of valid configurations with optimal codeword selection for effective TDV and TAT reduction. The PBO instance consists of two parts: Φ, $\mathcal{F}$. The formulation of the constraints Φ spans the solution space such that each solution to the set of constraints is a valid configuration and vice versa. The optimization function $\mathcal{F}$ is then used to rate each solution in terms of their costs. By this, the search is guided toward the most beneficial solution.

For this, constraints have to be generated such that the solving algorithm is able to consistently assign these variables. Overall, the problem instance including these constraints is defined by:

$$\Phi = \Phi_{\mathrm{ME}} \wedge \Phi_{\mathrm{uC}} \wedge \Phi_{\mathrm{RET}} \wedge \Phi_{\#\mathrm{CDW}}$$

Φ_{uC} realizes that all possible segments are covered to ensure the completeness, Φ_{ME} ensures that all bits of the original test data $\mathcal{I}$ are covered exactly once, $\Phi_{\#\mathrm{CDW}}$ guarantees that the maximum number of dictionary entries is not surpassed and, finally, Φ_{RET} implements the retargeting itself. A detailed description of the instance generation can be found at [19].

The constraints described so far restrict the solution space of the problem instance in a way that all valid configurations are part of the solution space and invalid configurations are no solutions. However, an optimization function is strictly necessary to ensure the effectiveness of the approach. If the problem is formulated as a decision problem, each valid configuration can potentially be chosen. Most likely, a solving algorithm would choose a solution, in which each bit position is covered by a single bit codeword, since this is a very easy solution to find. Therefore, the quality of a configuration has to be encoded to guide the solving algorithm to find a cost-effective solution.

The quality of a solution could be directly associated with the active segments in the data stream. However, in order to calculate the length of the compressed data stream accurately, three so-called cost variables have to be associated with each segment. Implications on the active segments and used codewords can be used to determine the specific length of the compressed stream. For improving the run time, the optimization function was only used based on the knowledge of the segment length, i.e., using existing variables.

The optimization function $\mathcal{F}$ is formulated over all segment variables $\nu_1^{uC}, \ldots, \nu_n^{uC}$. The weight of the variables depends on the number of bits the segment covers. Obviously, the more bits a segment covers, the better the solution and, consequently, the smaller the weight.

Beside this main optimization function $\mathcal{F}$, a secondary optimization function $\mathcal{F}_2$ is utilized and considered automatically during the optimization process. This function is used to determine the length of the CDW to be used for an active UDW, i.e., whether a codeword of length 2 or 3 is used. The main idea is that UDWs which occur more often are encoded by shorter codewords, i.e., of length 2. Since $\mathcal{F}_2$ has a lower priority, the result of $\mathcal{F}$ cannot be decreased, but the length of the CDW used for each active UDW is determined, i.e., 2 or 3, such that the costs are minimized. The run time overhead of using $\mathcal{F}_2$ is negligible. As an alternative, this can be also achieved in a post-processing step.

After the solution has been found, a configuration and the corresponding sequence of CDWs can be directly extracted from the variable assignment. Due to the high complexity of the invoked optimization-based techniques, a partitioning scheme has further been proposed in [20] allowing to process even very large industrial-scaled test data.

V. Hybrid Compression Architecture

Embedded compression techniques significantly reduce the test data volume, allowing coping even with the large test sets of highly complex IC designs. However, a certain share of the overall test patterns is rejected by the compression infrastructure leading to a shortcoming that jeopardizes the zero-defect policy for safety-critical applications.

This section proposes a novel hybrid compression architecture that tackles the before-mentioned drawback by combining a state-of-the-art embedded compression technique with a codeword-based approach, which is meant to be applied for the RTPs exclusively. By this, the significant overhead introduced by the RTPs can be effectively reduced while, at the same, keeping the fault coverage high.

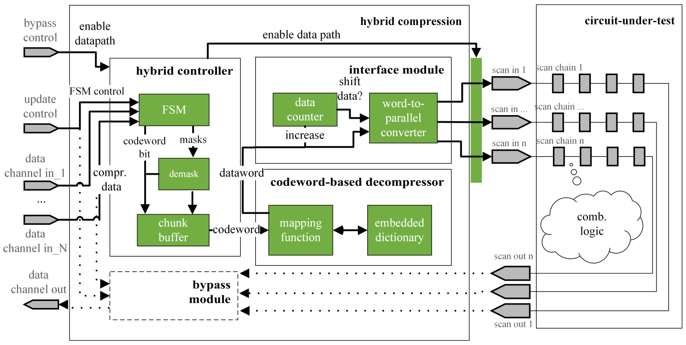

Figure 2: Hybrid Compression Architecture

The basic idea of the proposed hybrid compression architecture is about introducing a codeword-based decompressor instead of a simple bypass module as generally applied to address RTPs. The proposed scheme is shown in Figure 2 and focuses on the incoming-data of RTPs, which have to be retargeted once prior to the test application. The retargeted patterns, i.e., a sequence of codewords, are transferred bit-wise to the circuit. When a codeword is completed, the decompressor expands it to the (original) dataword and temporarily stores it until enough data are available to feed every input of the scan chains simultaneously. This is a significant improvement over the regular bypass since it can feed the test data into the scan chains in parallel and not serially as the regular bypass does.

Three components have to be implemented on-chip for instantiating the proposed hybrid compression technique as follows:

1) The *Codeword-based Decompressor* $\mathcal{D}_{CB}$ implements the embedded dictionary and provides an interface to the mapping function Ψ,
2) the *Hybrid Controller* $\mathcal{H}$ controls the operations of the codeword-based components, which includes the newly introduced control signal *hybrid_en* that configures the external datapath, and ensures that the hybrid compression components behave transparently if not activated and
3) the *Interface Module* $\mathcal{I}$ implements the junction between the newly introduced codeword-based decompressor and the inputs of the scan-chains.

The *Codeword-based Decompressor* implements the codeword-based technique utilizing an embedded dictionary to decompress the RTPs on-chip without any loss of information. Reconsider that these RTPs have been compressed once by the retargeting procedure prior to the test, i.e., the compressed data consists of a sequence of codewords $c_1, \dots, c_n$. A method to determine efficient codewords has been described in Section IV. The *Hybrid Controller* implements the necessary control structures by introducing a FSM, whose state transitions are controlled by the external compression control signal and synchronized with the test clock. By design, the FSM allows differentiating between data and instruction (*inst*) branch, which allows a clear distinction between data- and control-path. Besides this, another important criterion of the overall design of $\mathcal{H}$ is the datapath of $\mathcal{H}$, which remains completely isolated until the bypass control signal is set. This principle allows avoiding any unintended interference when the regular compression infrastructure decompresses the unrejected test patterns. Finally, the *Interface Module* realizes the junction between the introduced $\mathcal{D}_{CB}$ and the N inputs of the scan-chains. In this scenario, $\mathcal{D}_{CB}$ acts as the data source by decompressing newly received codewords to the associated datawords using Ψ and the inputs of the scan-chains act as the data sink. This module is especially important to feed the scan data into the scan chains in parallel, differently to the regular bypass, which works in a serial manner. As stated, the codewords of the embedded dictionary can be configured to arbitrary datawords (within the assumed boundary). Consequently, the length of the individual dataword is not necessarily the same, which ensures high flexibility and, thus, achieves high compression effectiveness. Due to this fact, a mechanism is required that keeps track of the actual number of available data bits, as described in detail in [21], [22].

VI. Enhanced Reliability using Formal Techniques

The general idea of the proposed methodology rests on the following observations: Today's circuits usually contain a vast number of FFs, which can store at least a single bit, i.e., '0' or '1'. If a single FF is affected by a transient fault, this bit is toggled. Existing approaches insert redundant logic into the design, e.g., to recompute the correct value, which causes a significant hardware overhead. At the same time, the value of an observed single FF is often equal to the value of many other FFs. Moreover, since the behavior of the circuit is known, it is possible to determine the relation between them, i.e., the states in which certain FFs assume the same value. Thus, instead of introducing redundancy for recomputations, we propose to simply compare the value of a FF to the values of other FFs from which it is known that, for the respectively considered state, they are supposed to generate the same value.

In order to realize this idea, a formalism is required that posts whether a partition of non-robust FFs assumes the same value for given reachable states. In the following, this is formally described in terms of an equivalence property.

Definition 2. *Let $P_j \subseteq N$ be a partition of at least two non-robust FFs and $\widehat{S} \subseteq S^*$ be a set of reachable states. Then, an* Equivalence Property *(*EP*) is defined by*

$$\mathsf{EP}(\widehat{S}, P_j) := \left\{ f_1, \dots, f_l \in P_j \;\middle|\; \begin{array}{l} \text{all FFs } f_1, \dots, f_l \text{ outputs the} \\ \text{same value under the same} \\ \text{state } s \in \widehat{S} \end{array} \right\}$$

and evaluates to true if all combinations of FFs $f_n, f_m \in P_j$ assume the same output value in all of these states $\widehat{S} \subset S^*$.

Example 3. *Consider the circuit shown in the upper part of Fig. 3, which is composed of five FFs distributed in two hierarchical circuit levels* 1 *and* 2*. If both FF_1 and FF_2 (level* 1*) are set to '0', then FF_3, FF_4, and FF_5 (level* 2*) are assumed to have the same output value '0' after a single clock cycle. This scenario is represented by an* $\mathsf{EP}(\widehat{S}, P_j)$ *with the partition $P_j = \{FF_3, FF_4, FF_5\}$ and the state $s_j \in \widehat{S}$ being defined by $FF_1 = 0$* and *$FF_2 = 0$, i.e.,* $\mathsf{EP}(\widehat{S}, P_j) = 1$ *holds.*

By taking advantage of the *Equivalence Property* [23] and the general idea sketched above, the robustness of sequential circuits is enhanced as follows:

1) Determine the set $N \subseteq \mathsf{SE}$ of non-robust FFs of the given sequential circuit. The assessment of robustness, as reviewed in Section II-C can be utilized.
2) Consider all non-robust FFs N and determine the level-wise subsets $N_i \cup N_{i+1} \cup \dots \cup N_L = N$ with $1 \leq i \leq L$ according to their hierarchical circuit levels (L being the total number of hierarchical levels in a rank-ordered circuit [42, p. 45]). Furthermore, assume that each FF has exactly one hierarchical level: $N_i \cap N_j = \emptyset \; \forall i \neq j$.

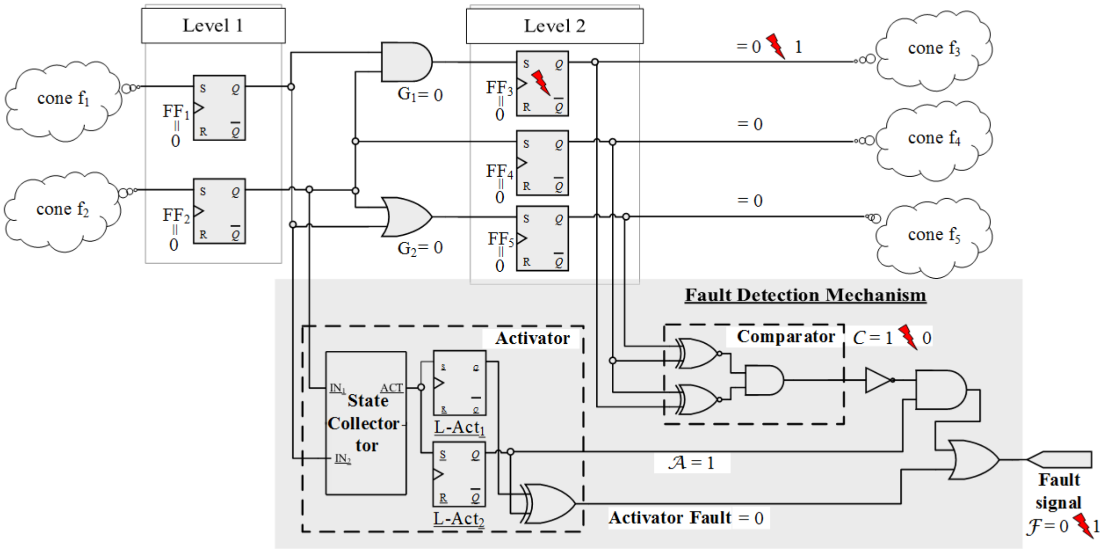

Figure 3: Applying the proposed methodology

The above-described clustering allows executing all FFs' comparisons within one single time frame. This is crucial to reduce the complexity of the calculation itself and, especially, the costs of the robustness improvement. Consequently, there is no need to hold specific FF values over different time frames, e.g., by introducing further, potentially vulnerable, state elements while massively increasing the computational effort as well as hardware scale. Thus, the level-wise sets of non-robust N_i are exclusively used in the remainder.

3) For each level $1 \leq i \leq L$ and for all subsets of non-robust FFs $N_i \subseteq N$, determine suitable partitions $P_j \in \mathcal{P}(N_i)$ and a set of reachable states $\widehat{S} \subseteq S^*$ such that all FFs in P_j are supposed to generate the same value, i.e., determine P_js and corresponding $\widehat{S}$s for which $\mathsf{EP}(\widehat{S}, P_j) = 1$ holds.
4) Using the knowledge from the obtained EPs, synthesize a FDM.

To this end, realize the following logic blocks:

1) *Activator* $\mathcal{A}$: Generates a signal $\mathcal{A}$ (supposed to trigger the FDM) stating whether ($\mathcal{A} = 1$) or not ($\mathcal{A} = 0$) the FFs in P_j are supposed to generate the same value under the current state $s \in \widehat{S}$. This signal is directly calculated by the current state (single time frame) of FFs within the fan-in cone. More precisely, it is not required to consider previous values, which is solely enabled by the hierarchical sort as described above in Step 2).
2) *Comparator* $\mathcal{C}$: Generates a signal $\mathcal{C}$ stating whether ($\mathcal{C} = 1$) or not ($\mathcal{C} = 0$) all FFs in a partition P_j to be hardened actually assume the same output value.
3) *Detector*: Generates a fault signal $\mathcal{F}$ reporting the detection of a fault. A fault is detected, if not all FFs in a partition P_j assume the same output value (i.e., $\mathcal{C} = 0$), although they are supposed to do that for the current state (i.e., $\mathcal{A} = 1$), i.e., $\mathcal{F} = \neg\mathcal{C} \wedge \mathcal{A}$.

This proposed FDM detects transient faults occurring in FFs of the considered circuit. If a fault is detected, an introduced fault signal $\mathcal{F}$ is driven. This enables the realization of precautions against faulty behavior at the POs, e.g., by resetting the circuit or masking the affected POs. Overall, this leads to enhanced robustness. The ratio of the enhancement can thereby be controlled, e.g., by adjusting the number knowledge collected through the EPs.

The *bottleneck* of the proposed methodology is the determination of – as much as possible – application-specific knowledge in terms of EPs. Ensuring the completeness would require that all possible partitions $P_j \in \mathcal{P}(N_i)$ of all non-robust FFs in the same hierarchical circuit level are considered. Obviously, this leads to an exponential complexity, which is not feasible for practical applications. Moreover, most of the partitions P_j are likely to be not suitable for an EP since no state s_j may exist for them so that all assume the same value.

In order to realize this proposed approach effectively, a mechanism is introduced, which aims to determine *good* partitions. Particularly, a SAT-based ATPG model is adopted to compute the criteria of quality for an investigated partition [24]. Besides this, a hardware-based evolutionary algorithm has recently been proposed in [25], which takes benefit of a at-speed partition enumeration and, hence, allow to determine highly effective partitions.

In addition to that, we heavily exploit formal methods such as BMC, powerful solvers for the SAT problem, and compact data structures involving *Binary Decision Diagrams* (BDDs, [43]). Eventually, this leads to a methodology composing:

1) A *Partition Enumerator* selects suitable partitions $P_j \in \mathcal{P}(N_i)$ which have not been considered before.
2) A *State Collector* determines the states $\widehat{S}$ under which all FFs in the selected partition P_j assume the same value and, hence, determines all $\mathsf{EP}(\widehat{S}, P_j)$ evaluating to true.
3) An *FDM Synthesizer* takes the obtained knowledge, realizes the FDM, and, eventually, embeds the resulting logic into the original circuit.

VII. Experimental Evaluation

This section presents the experimental evaluation of the three main contributions of this paper. The proposed TAP-controller with embedded compression and the hybrid compression architecture have been solely implemented in Verilog. Multiple *Verilog Parameters* has been introduced in the implementation to achieve a generic realization for varying bus sizes or the number of channels such that an easy adoption to further circuits is allowed. The generation of the considered test patterns and the simulation-based validation of the embedded compression have been performed by commercial tools. The retargeting runs and the BMC-based state analysis for the robustness enhancement have been executed on an *Intel(R) Core(TM) i7-7700 CPU @ 3.60GHz* with *32 GB* system memory within a *C++* compiler-environment (*gcc-Version 8.2.1*).

A. TAP-Controller with Embedded Compression

Several experiments have been conducted to show the superiority of the proposed approach, as presented in Table I and II: **leg**, the IEEE 1149.1 protocol without any compression at all forming the baseline for the evaluation, the **Huffman** retargeting procedure using a domain-specific Huffman algorithm [44] as previous work, i.e., the existing hardware constraints concerning the TDR interface are considered internally. Additionally, the **heur** algorithm invokes the greedy retargeting procedure as proposed by us in [18]. This approach allows a fast retargeting but does not determine an optimal set of codewords due to the greedy character yielding lower effectiveness of the compression.

Finally, **opt**, a retargeting procedure using formal techniques with exhaustive traversal and **opt-lim**, a modified version of **opt** limiting the computational effort to reduce the resulting run time significantly. These runs consider random test data with sizes from 2048 (RTDR_2048) to 8192 (RTDR_8192) bytes (generated by a pseudo-random number generator based on Mersenne Twister), commercially representative FV test cases of the *MiBench* benchmark suite [45], which have been cross-compiled for a state-of-the-art softcore microprocessor and *golden* signature data for a JPEG encoder circuit given as input debug data following the technique of [46]. Note that larger benchmarks have been considered in [20] by introducing a configuration-state aware partitioning scheme. These runs are excluded here due to the page limitation.

The results show that the proposed *opt* technique is able to improve both quality criteria – namely TDV and TAT reduction – for all test cases compared to the legacy TAP and previous techniques such as [44] significantly. Compared to the legacy TAP, the TDV can be reduced for high-entropy random data on average by 36.4% and up to 37.1%. The application on the debugging data or on FV data confirms these results. Here, the proposed technique is able to reduce the TDV by 47.6%. It is also shown that a large run time benefit can be achieved if the resources of the solving algorithm are limited (*opt-lim*), which decreases the compression ratio only slightly. The proposed approach does not suffer this circumstance due to the use of formal optimization techniques. It is able to even reduce the data cycles by 2.2% on average and up to 2.9% for random high-entropy data and by 15% on average and up to 20.0% for the debug or FV data.

B. Hybrid Compression Architecture

Table III presents the detailed results of the hybrid architecture for the conducted experiments. More precisely, the benchmark circuit name, the overall number of test patterns, which have been generated by the commercial tool (#pattern), the number of required data blocks per RTP, the minimal, average and maximal retargeting run-time per pattern in seconds are shown. Furthermore, Table III also presents the minimal, average and maximal pattern compression ratio in percent and, finally, the further achieved test time reduction in percent - both compared against the regular bypass transfer without any compression at all - and, finally, the minimal, average and maximal achievable test time (volume) reduction in percent when considering the multichannel interface – if available and, otherwise, a 'n/a' is shown.

The proposed approach is complete by construction, i.e., any arbitrary RTP can be transferred and, thus, no test coverage loss is introduced. In case of the *netcard* benchmark, at first, the experiments clearly show that the run-time of the retargeting engine, while considering different design sizes, is stable **per** block. Since the retargeting has been applied only once after the test pattern generation process, the run-time is manageable and, furthermore, the retargeting invokes currently only one single thread and the individual blocks are objects to be parallelized.

Besides this, the achieved compression ratio is stable over the conducted experiments as well, which is indicated by a variance standard deviation of 6.4% (*netcard*). In contrast to this, the deviation between the minimum and the maximum of achieved test time reduction is greater, which is due to the fact that this depends on the distribution of codewords with a length of 1, 2 or 3, respectively. Thus, the test time reduction is not directly connected with the compression ratio. Latter is determined mainly by the associated datawords the introduced codewords are pointing to. When processing the largest *netcard* circuit with the greatest number of patterns ($N = 9939$), the resulting test data volume can be significantly compressed by 38.1% on average in conjunction with a test time reduction of 45.7% on average. By invoking the proposed multichannel scheme, the test application time can be further reduced by up to 72.9% while still achieving a compression ratio up to 48.9%, which exhibits a large potential of saving test costs.

C. Robustness Enhancement

The proposed methodology has been implemented in C++. For determining the non-robust FFs of the circuit, a simulation-based robustness checker has been implemented which transforms the given circuits (provided in *Verilog* and parsed by *Verific*) into a compiled simulation model (to this end, *LLVM* [47] *IR code* is generated by the simulation environment). In order to conduct the respective BMC task, *MiniSAT* on top of *metaSMT* [48], together with the *X-value abstraction* as described in [49] has been utilized. The BDD package *CUDD* has been used to generate the MUX circuits. Afterwards, the resulting flow has been evaluated using *ITC'99* benchmark circuits. In order to determine the set of all non-robust FFs.

Figure 4 shows the robustness of the original circuit as well as the robustness after applying the proposed methodology (for different partition sizes $p_s \in \{4, 8, 16\}$). The proposed methodology provides a suitable alternative to previously proposed solutions, such as discussed in Section I. Although space-based approaches such as TMR [13] can guarantee 100% robustness, they usually require more than thrice the amount of hardware (i.e., yielding a scaling factor of > 3.0). In contrast, the solution proposed in this work is capable of always improving the robustness to more than 90% (in some cases even close to 100%), while only a factor of approx. 1.13 more hardware is required for this on average. As already discussed before, the proposed solution also outperforms timing-based and application-specific approaches, since timing is hardly affected at all in the proposed solution and the methodology can be applied to arbitrary sequential circuits. By this, a suitable trade-off between enhancing the robustness and keeping the hardware overhead small is achieved.

TABLE I: Benchmarks: Processing random test data & debug data considering TDV

No.	test name	run time [min]				size [bit]					data reduction [%]			
		Huffman	heur	opt	opt-lim	leg	Huffman	heur	opt	opt-lim	Huffman	heur	opt	opt-lim
1	RTDR_2048	0.05	0.19	97.59	0.76	16384	12973	12109	10444	10739	20.8	26.1	36.3	34.5
2	RTDR_4096	0.10	0.63	190.89	14.85	32768	26005	23607	20626	21282	20.6	28.0	37.1	35.1
3	RTDR_8192	0.22	2.16	492.08	29.47	65536	51119	48529	42068	43057	22.0	26.0	35.8	34.3
4	SHA	0.06	0.10	217.10	37.49	46944	34861	28154	23271	23271	25.7	40.1	50.4	50.4
5	MATH	0.09	0.15	314.28	68.42	67616	57470	48217	33876	35220	15.0	28.7	49.9	47.9
6	DIJKSTRA	0.06	0.10	183.78	38.76	49441	34056	29172	24621	25165	31.2	40.1	50.2	49.1
7	FFT	0.08	0.14	356.07	72.11	66880	48856	39743	34034	34310	26.9	40.6	49.1	48.7
8	DEBUG_JPEG	< 0.01	0.02	30.86	2.13	5632	4134	4069	3470	3470	26.6	27.8	38.4	38.4

TABLE II: Benchmarks: Processing random test data & debug data considering TAT

No.	test name	#Boolean variables	#data-cycles					TAT reduction [%]			
		opt/opt-lim	leg	Huffman	heur	opt	opt-lim	Huffman	heur	opt	opt-lim
1	RTDR_2048	58930	16389	20603	18013	16082	16378	-25.7	-9.9	1.9	1.0
2	RTDR_4096	117203	32773	41442	35236	31809	32591	-26.5	-7.5	2.9	0.5
3	RTDR_8192	233997	65568	81951	71847	64420	64494	-25.0	-9.6	1.7	1.5
4	SHA	204388	46949	51921	47090	37561	37561	-10.6	-0.3	20.0	20.0
5	MATH	286954	67621	84851	74253	57152	57264	-25.5	-9.8	15.4	15.3
6	DIJKSTRA	222101	49446	55949	49493	40533	41210	-13.2	-0.1	18.0	16.7
7	FFT	289219	66885	79393	67027	55345	55733	-18.7	-0.2	17.3	16.7
8	DEBUG_JPEG	21449	5637	6871	6125	5412	5412	-27.0	-13.2	4.0	4.0

TABLE III: Hybrid compression benchmark results

circuit name	#pattern	#blocks	retargeting run-time [s]			pattern compression ratio [%]			test time red. [%]			multichannel test time (volume) red. [%]		
			min.	avg.	max.	min.	avg.	max.	min.	avg.	max.	min.	avg.	max.
ethernet	1,049	2	13.2	26.4	47.2	31.1	56.4	79.3	34.6	55.5	72.8	n/a	n/a	n/a
vga_lcd	1,286	2	13.6	37.6	42.1	30.6	40.7	56.1	30.5	41.5	47.9	n/a	n/a	n/a
leon3	6,332	11	64.3	293.7	513.2	31.7	48.7	79.2	27.2	48.8	69.4	38.4 (14.8)	59.4 (48.1)	84.8 (77,9)
netcard	9,939	12	104.4	126.0	230.4	28.5	38.1	67.4	29.6	45.7	65.7	34.6 (20.1)	51.2 (39.8)	72.9 (48.9)

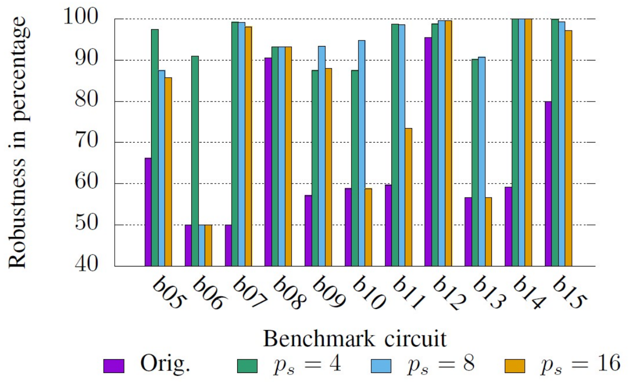

Figure 4: Robustness for circuits with different partition sizes

VIII. Conclusion & Outlook

This work proposed several novel approaches for tackling some of the rising challenges in the field of design for testability, debug, and reliability, as strictly required for state-of-the-art circuit designs. In particular, this work combines formal techniques, such as the Boolean satisfiability problem and the bounded model checking to significantly reduce the resulting test data volume and test application time and strongly enhance the reliability of the circuit against transient faults. The proposed measures have been integrated into a common framework, which implements standardized software/hardware interfaces, and, hence, can be reused in an industrial environment that uses a commercial electronic design automation tool-flow.

Experiments have shown that the proposed **TAP-controller with embedded compression** – in combination with the optimization-based retargeting framework – allows significantly reducing the TDV and TAT and, by this, outperforming other existing techniques. In fact, this technique reduces the TDV for FV data by up to 50.4%. The TDV is even reduced for high-entropy test and debug data by up to 37.1%. Furthermore, the TAT is also reduced by up to 20.0% compared to the TAT of a legacy transfer while processing FV data.

Different experiments on industrial-representative designs – with up to approximately 100k flip-flops – demonstrate the effectiveness of the proposed **hybrid architecture**. A significant test data volume reduction of up to 67.4% was achieved and, most importantly, without any loss in test coverage at all. Furthermore, the test application time was reduced even stronger by up to 64.7%. In addition to this, the proposed hybrid architecture allows utilizing existing multichannel structures. By this, it was possible to further decrease the test application time by up to 72.9% with up to 48.9% test data volume reduction.

The proposed **robustness enhancement** technique allows exploiting application-specific knowledge about the FFs in each reachable state. To this end, a methodology is introduced, which gains the corresponding knowledge and, afterward, utilizes them for a fault detection mechanism. To cope with the underlying complexity, a dedicated orchestration of formal techniques is employed. This results in a hardening method requiring only a slight increase in additional hardware, does only influence the timing behavior negligibly by introducing just one further fan-out to the FFs, and is automatically applicable to arbitrary circuits. Experimental evaluations confirmed these benefits: The robustness can be increased to approx. 84% (97%), while the circuit size increases only by a factor of approx. 1.07 (1.07) on avg. while applying the random-based (SAT-based) approach.

Future work will focus on the test scheduling problem in large IJTAG test networks with multiple power domains by orchestrating formal techniques. Initial works have been presented in [26], [50]. Besides this, future work will investigate introducing machine learning techniques on-chip to implement an on-chip compressor, as preliminary proposed in [51].

IX. Acknowledgment

The authors would like to thank Dr. Stephan Eggersglüß, Dr. Stefan Frehse, Prof. Daniel Große, Dr. Matthias Sauer, and Prof. Robert Wille for various inspiring conversations and helpful support. Furthermore, the authors thank Infineon Germany AG and, in particular, Dr. Daniel Tille for the productive cooperation. We would like to thank Verific Design Automation Inc. for providing the SystemVerilog frontend used for the implementation of our technique. Parts of this research work were supported by the AI initiative of the Free Hanseatic City of Bremen and by the German Research Foundation (DFG) in terms of the Collaborative Research Center 1232.

References

[1] Y. Zorian and S. Shoukourian, "Embedded-memory test and repair: infrastructure IP for SoC yield," *IEEE Transaction on Design Test of Computers*, vol. 20, no. 3, pp. 58–66, 2003.

[2] S. Eggersglüß, S. Holst, D. Tille, K. Miyase, and X. Wen, "Formal test point insertion for region-based low-capture-power compact at-speed scan test," in *IEEE Asian Test Symposium*, 2016, pp. 173–178.

[3] J. Rajski, J. Tyszer, M. Kassab, and N. Mukherjee, "Embedded deterministic test," *IEEE Transactions on VLSI Systems*, vol. 23, no. 5, pp. 776–792, 2004.

[4] A. Jas and N. A. Touba, "Test vector decompression via cyclical scan chains and its application to testing core-based designs," in *International Test Conference*, 1998, pp. 458–464.

[5] F. G. Wolff and C. Papachristou, "Multiscan-based test compression and hardware decompression using LZ77," in *International Test Conference*, 2002, pp. 331–339.

[6] V. Iyengar, K. Chakrabarty, and B. T. Murray, "Deterministic built-in pattern generation for sequential circuits," *Journal of Electronic Testing*, vol. 15, no. 1-2, pp. 97–114, 1999.

[7] A. Jas, J. Ghosh-Dastidar, and N. A. Touba, "Scan vector compression/decompression using statistical coding," in *VLSI Test Symposium*, 1999, pp. 114–120.

[8] A. Chandra and K. Chakrabarty, "System-on-a-chip test-data compression and decompression architectures based on Golomb codes," *IEEE Transactions on VLSI Systems*, vol. 20, no. 3, pp. 355–368, 2001.

[9] A. Wurtenberger, C. S. Tautermann, and S. Hellebrand, "Data compression for multiple scan chains using dictionaries with corrections," in *International Test Conference*, 2004, pp. 926–935.

[10] S. Mitra and K. S. Kim, "XPAND: an efficient test stimulus compression technique," *IEEE Transactions on Computers*, vol. 55, no. 2, pp. 163–173, 2006.

[11] R. Jia, F. Wang, R. Chen, X.-G. Wang, and H.-G. Yang, "JTAG-based bitstream compression for FPGA configuration," in *Proceedings of the International Conference on Solid-State and Integrated Circuit Technology*, 2012, pp. 1–3.

[12] A. E. Barbour and A. S. Wojcik, "A general constructive approach to fault-tolerant design using redundancy," *IEEE Transactions on Computers*, vol. 38, no. 1, pp. 15–29, 1989.

[13] C. E. Stroud and A. E. Barbour, "Design for testability and test generation for static redundancy system level fault-tolerant circuits," in *International Test Conference*, 1989, pp. 812–818.

[14] D. Ernst, N. S. Kim, S. Das, S. Pant, R. Rao, T. Pham, C. Ziesler, D. Blaauw, T. Austin, K. Flautner, and T. Mudge, "Razor: a low-power pipeline based on circuit-level timing speculation," in *Proceedings of the IEEE/ACM International Symposium on Microarchitecture*, 2003, pp. 7–18.

[15] D. Blaauw, S. Kalaiselvan, K. Lai, W. H. Ma, S. Pant, C. Tokunaga, S. Das, and D. Bull, "Razor II: in situ error detection and correction for PVT and SER tolerance," in *IEEE International Conference on Solid-State Circuits*, 2008, pp. 400–622.

[16] S. A. Cook, "The complexity of theorem-proving procedures," in *ACM International Symposium on the Theory of Computing*, 1971, pp. 151–158.

[17] A. Biere, A. Cimatti, E. Clarke, and Y. Zhu, "Symbolic model checking without BDDs," in *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 1999, pp. 193–207.

[18] **S. Huhn**, S. Eggersglüß, and R. Drechsler, "VecTHOR: Low-cost compression architecture for IEEE-1149.1-compliant TAP controllers," in *IEEE European Test Symposium*, 2016, pp. 1–6.

[19] **S. Huhn**, S. Eggersglüß, K. Chakrabarty, and R. Drechsler, "Optimization of retargeting for IEEE 1149.1 TAP controllers with embedded compression," in *Design, Automation and Test in Europe*, 2017, pp. 578–583.

[20] **S. Huhn**, S. Eggersglüß, and R. Drechsler, "Reconfigurable TAP controllers with embedded compression for large test data volume," in *IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, 2017, pp. 1–6.

[21] **S. Huhn**, D. Tille, and R. Drechsler, "Hybrid architecture for embedded test compression to process rejected test patterns," in *IEEE European Test Symposium*, 2019, pp. 1–2.

[22] **S. Huhn** and D. Tille and R. Drechsler, "A hybrid embedded multichannel test compression architecture for low-pin count test environments in safety-critical systems," in *International Test Conference in Asia*, 2019, pp. 115–120.

[23] **S. Huhn**, S. Frehse, R. Wille, and R. Drechsler, "Enhancing robustness of sequential circuits using application-specific knowledge and formal methods," in *ASP Design Automation Conference*, 2017, pp. 182–187.

[24] **S. Huhn** and S. Frehse, R. Wille, and R. Drechsler, "Determining application-specific knowledge for improving robustness of sequential circuits," *IEEE Transactions on VLSI Systems*, pp. 875–887, 2019.

[25] M. Merten, **S. Huhn**, and R. Drechsler, "A hardware-based evolutionary algorithm with multi-objective optimization operators for on-chip transient fault detection," in *VLSI Test Symposium*, 2022, pp. 1–6.

[26] P. Habiby, **S. Huhn**, and R. Drechsler, "Power-aware test scheduling framework for IEEE 1687 multi-power domain networks using formal techniques," *Microelectronics Reliability*, vol. 134, p. 114551, 2022.

[27] **S. Huhn** and R. Drechsler, *Next Generation Design For Testability, Debug and Reliability Using Formal Techniques*. Springer, 2021, 1–185.

[28] V. Iyengar, S. K. Goel, E. J. Marinissen, and K. Chakrabarty, "Test resource optimization for multi-site testing of SOCs under ATE memory depth constraints," in *International Test Conference*, 2002, pp. 1159–1168.

[29] T. Heijmen and A. Nieuwland, "Soft-error rate testing of deep-submicron integrated circuits," in *IEEE European Test Symposium*, 2006, pp. 247–252.

[30] R. C. Baumann, "Radiation-induced soft errors in advanced semiconductor technologies," *IEEE Transaction on Device and Materials Reliability*, vol. 5, no. 3, pp. 305–316, 2005.

[31] U. Krautz, M. Pflanz, C. Jacobi, H. W. Tast, K. Weber, and H. T. Vierhaus, "Evaluating coverage of error detection logic for soft errors using formal methods," in *Design, Automation and Test in Europe*, 2006, pp. 1–6.

[32] L. Doyen, T. A. Henzinger, A. Legay, and D. Nickovic, "Robustness of sequential circuits," in *Proceedings of the International Conference on Application of Concurrency to System Design*, 2010, pp. 77–84.

[33] G. Fey, A. Sülflow, S. Frehse, and R. Drechsler, "Effective robustness analysis using bounded model checking techniques," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 8, pp. 1239–1252, 2011.

[34] G. Fey and R. Drechsler, "A basis for formal robustness checking," in *International Symposium on Quality Electronic Design*, 2008, pp. 784–789.

[35] S.-Y. Huang, K.-T. Cheng, K.-C. Chen, and J. Y. J. Lu, "Fault-simulation based design error diagnosis for sequential circuits," in *Design Automation Conference*, 1998, pp. 632–637.

[36] N. Miskov-Zivanov and D. Marculescu, "Multiple transient faults in combinational and sequential circuits: A systematic approach," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, no. 10, pp. 1614–1627, 2010.

[37] M. Bozzano, A. Cimatti, and F. Tapparo, *Symbolic Fault Tree Analysis for Reactive Systems*. Springer, 2007, ch. International ConferenceAutomated Technology for Verification and Analysis, pp. 162–176.

[38] D. Nayak and D. M. H. Walker, "Simulation-based design error diagnosis and correction in combinational digital circuits," in *VLSI Test Symposium*, 1999, pp. 70–78.

[39] A. Biere, M. Heule, H. Maaren, and T. Walsh, *Handbook of Satisfiability*, ser. Frontiers in AI and Applications. IOS Press, 2009, vol. 185.

[40] S. Eggersglüß, R. Wille, and R. Drechsler, "Improved SAT-based ATPG: More constraints, better compaction," in *International Conference on Computer-Aided Design*, 2013, pp. 85–90.

[41] M. Sauer, B. Becker, and I. Polian, "PHAETON: A SAT-based framework for timing-aware path sensitization," *IEEE Transactions on Computers*, vol. 65, no. 6, pp. 1869–1881, 2016.

[42] A. Miczo, *Digital Logic Testing and Simulation*. Wiley-Interscience, 2003, vol. 2.

[43] R. E. Bryant, "Graph-based algorithms for boolean function manipulation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. C-35, no. 8, pp. 677–691, 1986.

[44] K. Ilambharathi, G. S. N. V. V. Manik, N. Sadagopan, and B. Sivaselvan, "Domain specific hierarchical Huffman encoding," *Cornell University Library*, vol. abs/1307.0920, 2013.

[45] M. R. Guthaus, J. S. Ringenberg, D. Ernst, T. M. Austin, T. Mudge, and R. B. Brown, "MiBench: A free, commercially representative embedded benchmark suite," in *Proceedings of the IEEE International Workshop on Workload Characterization*, 2001, pp. 3–14.

[46] S. Deutsch and K. Chakrabarty, "Massive signal tracing using on-chip DRAM for in-system silicon debug," in *International Test Conference*, 2014, pp. 1–10.

[47] C. Lattner and V. Adve, "LLVM: a compilation framework for lifelong program analysis transformation," in *International Symposium on Code Generation and Optimization*, 2004, pp. 75–86.

[48] H. Riener, F. Haedicke, S. Frehse, M. Soeken, D. Große, R. Drechsler, and G. Fey, "metaSMT: focus on your application and not on solver integration," *International Journal on Software Tools for Technology Transfer*, pp. 1–17, 2016.

[49] O. Grumberg, "3-valued abstraction for (bounded) model checking," in *International ConferenceAutomated Technology for Verification and Analysis*. Springer, 2009, pp. 21–21.

[50] P. Habiby, **S. Huhn**, and R. Drechsler, "Optimization-based test scheduling for IEEE 1687 multi-power domain networks using boolean satisfiability," in *IEEE Design & Technology of Integrated Systems in Nanoscale Era*, 2021, pp. 1–6.

[51] M. Merten, **S. Huhn**, and R. Drechsler, "A codeword-based compactor for on-chip generated debug data using two-stage artificial neural networks," in *IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, 2021, pp. 1–6.