A Hybrid Protection Scheme for Reconfigurable Scan Networks

Lylina, Natalia; Atteya, Ahmed; Wunderlich, Hans-Joachim

Proceedings of the IEEE VLSI Test Symposium (VTS'21), Virtual, 25 - 28 April 2021, pp. 1-7

doi: https://doi.org/10.1109/VTS50974.2021.9441029

Abstract: The reliable operation of integrated systems is supported by Reconfigurable Scan Networks (RSNs) which allow to access efficiently the embedded instruments throughout the lifecycle. However, the RSN integration may introduce additional connectivities into a Device-under-Test (DUT), and the RSN might be misused for information leakage. Structural methods resynthesize the RSNs and add hardware components such that certain instruments are physically separated, while functional approaches add filters to prevent certain access patterns. Both methods have certain limitations.This paper presents an effective approach to maximize the benefits and to overcome the limitations of the existing solutions by a hybrid combination of structural and functional protection schemes. A minimized number of structural changes is identified in order to resolve violations which cannot be handled by using sequence filters. The remaining violations are resolved functionally by using filters and a flexible protection can be enabled for multiple user groups with different access permissions. Since the majority of the violations are resolved using a filter, the hardware overhead for structural changes is drastically reduced. The efficiency of the approach is supported by experimental results.

Preprint

General Copyright Notice

This article may be used for research, teaching and private study purposes. Any substantial or systematic reproduction, re-distribution, re-selling, loan or sub-licensing, systematic supply or distribution in any form to anyone is expressly forbidden.

This is the author's "personal copy" of the final, accepted version of the paper published by IEEE.[1]

[1] **IEEE COPYRIGHT NOTICE**

©2021 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

A Hybrid Protection Scheme for Reconfigurable Scan Networks

Natalia Lylina, Ahmed Atteya and Hans-Joachim Wunderlich
ITI, University of Stuttgart, Pfaffenwaldring 47, D-70569 Stuttgart, Germany
{lylina, atteyaad, wu}@informatik.uni-stuttgart.de

***Abstract*—The reliable operation of integrated systems is supported by Reconfigurable Scan Networks (RSNs) which allow to access efficiently the embedded instruments throughout the life-cycle. However, the RSN integration may introduce additional connectivities into a Device-under-Test (DUT), and the RSN might be misused for information leakage. Structural methods resynthesize the RSNs and add hardware components such that certain instruments are physically separated, while functional approaches add filters to prevent certain access patterns. Both methods have certain limitations.**

This paper presents an effective approach to maximize the benefits and to overcome the limitations of the existing solutions by a hybrid combination of structural and functional protection schemes. A minimized number of structural changes is identified in order to resolve violations which cannot be handled by using sequence filters. The remaining violations are resolved functionally by using filters and a flexible protection can be enabled for multiple user groups with different access permissions. Since the majority of the violations are resolved using a filter, the hardware overhead for structural changes is drastically reduced. The efficiency of the approach is supported by experimental results.

***Keywords*-Reconfigurable Scan Networks, Information Leakage, Security Compliance, Design-for-Test**

I. Introduction

Reconfigurable Scan Networks (RSN), as standardized by IEEE Std. 1687 and 1149.1 [1, 2], support efficient reliability management [3] by providing access to the embedded instruments and controlling their operation. These instruments include Built-In Self-Test registers, aging monitors and sensors. The trustworthiness of the Intellectual Property (IP) cores of the DUT, data confidentiality and the access rights for various user groups can be explicitly specified by a system designer [4]. These properties together with the implicit security specification, determined by the connectivities inside the DUT, are referred as the *allowed information flow* and must be considered by scan chain integration. An improper scan chain integration [5–7], may extend the allowed information flow of the DUT, sacrifice the designer's efforts towards system-level security, and may be exploited to leak critical data or alter the system behavior, e.g. [8–11]. Fine-grained access management mechanisms, such as Locking Segment Insertion Bits (SIBs) [12, 13] or Secure SIBs [14], have been proposed to complicate an unauthorized access to specific RSN parts. In [11], the initial RSN is augmented with additional registers and control logic to improve the access trustworthiness. However, none of those approaches generally exclude the extension of the allowed information flow.

In [15] an accurate approach is presented to verify the compliance of a given RSN with the allowed information flow of the DUT and to identify the violating connectivities in the RSN. To prevent information leakage, these violating connectivities must be precluded either functionally, by restricting the set of test sequences using filters, or structurally, by resynthesizing some parts of the RSN.

1) Filter-based protection: Filters of access patterns can allow just a static set of precomputed accesses, as in [16], or provide access protection for complex access scenarios, as in [17]. Benefits of the filter-based approach are:

- Minimal hardware overhead nearly independent of the RSN complexity.
- Compliant with extensions of the RSN standards like the P1687.1 proposal [18] which defines access to RSNs through alternate interfaces.
- It is flexible and programmable, and can be adopted for changing security requirements.
- Together with a standard authorization scheme, it can handle different access rights for different user groups.

A severe drawback of the filter approach is the fact that there may exist security violations which cannot be resolved without unwanted side-effects. Fig. 1 shows an example, where any filter approach would sacrifice the accessibility of other instruments as well.

2) Resynthesis of the RSN Structure: Structural resynthesis as presented in [19–21] can resolve all security violations, but it comes with certain drawbacks as well:

- In some cases, major hardware costs are incurred even when applying sophisticated synthesis procedures [21].
- Retargeting patterns have to be recomputed.
- If security requirements are changing, a complete resynthesis is necessary.
- User group specific access rights cannot be given.

3) Hybrid protection scheme: While structural resynthesis can resolve all conflicts, it lacks flexibility. On the other hand, filters may sacrifice the required accessibility. The paper at hand avoids the drawbacks, its main contributions are:

- **Efficient analysis**: A filter applicability analysis (FAA) is presented to identify a minimized subset of violations, which cannot be handled using filters, while preserving the accessibility of the instruments through the RSN.
- **Minimized hardware overhead**: The structural changes are applied to resolve only this small subset and to ensure that any further violation is resolvable using a filter.

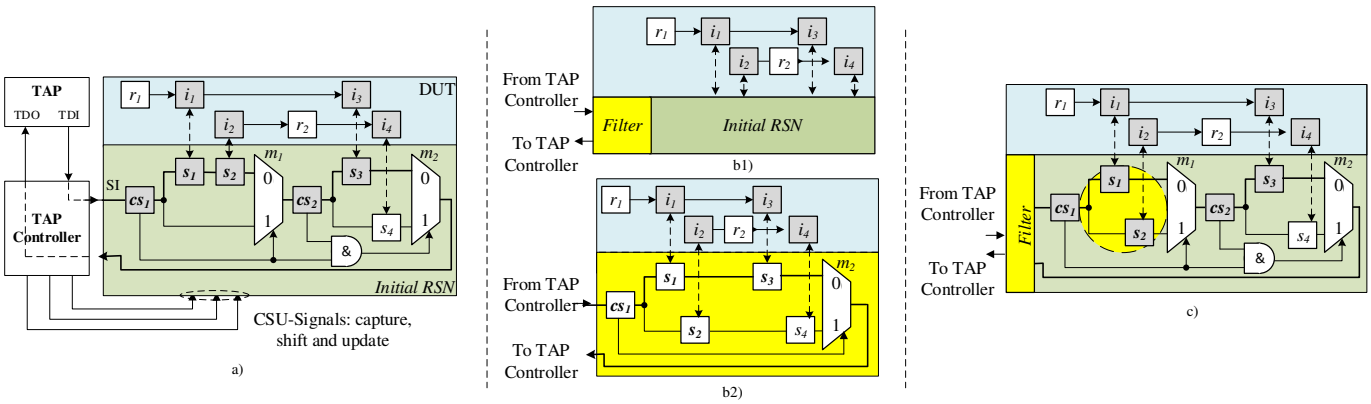

Fig. 1: a) The *DUT* includes the functional registers $r_j \in R$, the instruments' registers $i_j \in I$, and the connections between them. The *RSN* accesses the instruments' registers using a Test Access Port (TAP) interface through the segments $s_k \in S$. The connectivity between s_1 and s_2 introduces an additional connectivity from i_1 to i_2 through the RSN and an accessibility conflict arises. The same applies to the connectivities from s_1 to s_4 and from s_2 to s_3.
b1) If access to a safety-critical instrument i_1 is required for all user groups, using the *filter* is not an option: it makes s_1, s_2 and thereby i_1, i_2 inaccessible through the RSN. All the paths traversing s_1 also traverse s_2, and restricting an access to one of them implies a restriction for the second one.
b2) *Resynthesis* resolves this violation, preserving the accessibility, but does not consider various access permissions. If an access to i_3 is restricted for regular users, but not for test engineers, the RSN must be resynthesized again, implying even more HW costs.
c) *Hybrid Scheme* considers various access permissions and the required accessibility is preserved. The RSN structure is slightly modified compared to a), in order to make s_1 and s_2 accessible individually and thereby to ensure the instruments' accessibility. The remaining violations are resolved by the filter.

- **Flexible access**: Most of the violations are resolved online by using a flexible filter, to ensure access to RSNs for various user groups with different access permissions.
- **Preserved accessibility**: All the required instruments remain accessible through the RSN for all user groups.

The remainder of the paper is organized as follows. In Section II basic definitions are introduced. In Section III the necessary background on the RSN analysis and the protection schemes is given and serves as a basis for the presented approach. Section IV presents the filter applicability analysis. In Section V an overview of the hybrid protection scheme is given. Section VI provides the experimental results, which show the scalability of the presented method.

II. Definitions and Modeling

A. Reconfigurable Scan Network

An RSN (Fig. 1) is built using the following scan primitives:

- Scan Segments $s_j \in S$ access the instruments $i_j \in I$. Each scan segment consists of a shift register and an optional shadow register.
- Configuration Scan Primitives, such as scan multiplexers $m_j \in M$ and SIBs, are used to configure the paths in the RSN. The scan multiplexers select the specific branches to be included into a path, and the SIBs include or exclude certain parts of the RSN from the path.
- Control signals can be external of the RSN, or internal, coming from a shadow register of a control segment $cs_j \in S$. They drive the address ports of the configuration scan primitives and the select signals of the segments.

A connected path from a primary scan-in (SI) to a primary scan-out (SO) through a sequence of selected scan primitives is called an Active Scan Path (ASP). In Fig. 1, an initial ASP is SI - cs_1 - s_1 - s_2 - cs_2 - s_3 - SO. An access to an RSN can be represented as a *capture*, *shift* and *update* (CSU)-operation, as presented in [22]. The phases of a CSU-operation are controlled by the TAP controller, driving the external control signals to the RSN. The input data from the Test Data Input (TDI) of the TAP goes to the primary SI of the RSN and, after performing access, the data from the SO returns to Test Data Output (TDO). During the *capture*-phase the data from the instruments is read to the scan segments. Then, during the *shift*-phase the new data is being shifted-in through the SI and the existing data is being shifted-out. During the *update*-phase the shifted-in data is written to the shadow registers of the scan segments. This data can be sent to the instruments or used to generate internal control signals.

The state of the sequential elements defines the current scan configuration $c \in C$. According to IEEE Std. 1687 [1], in a valid scan configuration only one ASP is selected. The transition relation $T \subset C^2$ defines all the pairs of the scan configurations (c_1, c_2), such that c_2 is reachable from c_1 within a single CSU-operation. A sequence of CSU-operations can be required to form the desired ASPs, including the specific scan primitives, and transport the data. The computation of the control patterns for such a sequence is called retargeting.

B. Modeling

The system is modeled as a directed graph $G := (V, E)$ with the vertex set V and the edge set E as shown in Fig. 2. It consists of two sub-graphs, namely the DUT graph $G^{DUT} := (V^{DUT}, E^{DUT})$, and the RSN graph $G^{RSN} := (V^{RSN}, E^{RSN})$, shown in the upper and in the lower part.

The edge set E includes the connections inside the sub-graphs and between them.

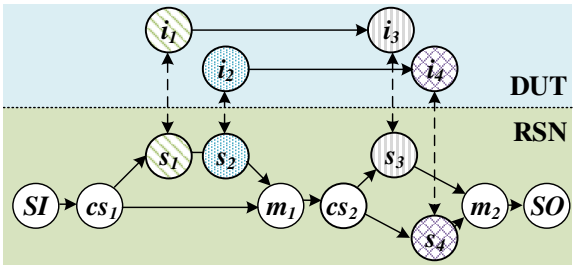

Fig. 2: System graph

The vertex set of the RSN graph includes the scan segments s_j and multiplexers m_j, primary SI and SO. The edge set of the RSN represents the direct connectivity between the vertex pairs. The vertices of the DUT graph represent the instruments I. The edges E^{DUT} represent the direct connectivities between the vertices or the connectivity through the functional registers. (e.g., an edge from i_2 to i_4 shows that i_4 is reachable from i_2 through r_2 in Fig. 1).

III. Background

A. Security Compliance Analysis (SCA) of RSNs

This subsection briefly summarizes the SCA of [15], which serves as an input for the hybrid protection scheme. The security properties of the DUT define the allowed information flow using implicit and explicit security specifications. An *implicit specification* is defined by the functional connectivity of the registers inside the DUT. The functional reachability computation can be accomplished, e.g., by means of false-path analysis [23, 24], information flow tracking [25] or SAT-based approaches [26], and lays outside the scope of this paper. Additional requirements are provided in the *explicit specifications* [4], which consider the designer's intentions on the system-level security. E.g., access or observation through specific functional or non-functional channels can be restricted, and the access rights might be specified to various user groups, starting from the normal users, upto the test engineers. For a user u_j a subset of segments $G(u_j) \subset S$ can be specified, such that u_j must not access $\forall s_k \in G(u_j)$. The subset $A(u_j)$ defines all segments, which must remain accessible for u_j.

The reachability analysis of an RSN identifies all the functional connectivities between those segments, which are used to access the instruments, and is divided into four major steps:

- All the structural connectivities in the RSN are determined by applying a transition relation over G^{RSN}.
- The control signals in the RSN are analyzed and the subset of connectivities, belonging to single valid scan configurations, is identified.
- The functionally possible connectivities, considering an unbounded sequence of scan configurations, are unrolled from the connectivities within single configurations.
- The connections between the instruments of the DUT through the scan segments of the RSN are computed.

Paths, traversing both the DUT and the RSN, are computed in order to define the connectivity of the instruments and the segments after the RSN integration. The security violations, such as authorization and security compliance violations, are identified for the user u_l and the scan segments s_j, s_k:

Definition 1: A *security compliance violation* $viol_{j,k}$ is a connectivity from the source s_j to the destination s_k, which extends the functional connectivity of the DUT or violates the explicit specification for at least one user u_l.

Definition 2: An *authorization violation* $viol_k$ is an unfulfilled requirement to restrict an access to $s_k \in G(u_l)$ for u_l.

B. Sequence Filter: Functional Changes

The sequence filter, as presented in [17], is a flexible way to resolve the violations online. It is constructed based on the RSN structural description and its specification, and is put between the TAP controller and the RSN, as shown in Fig. 3.

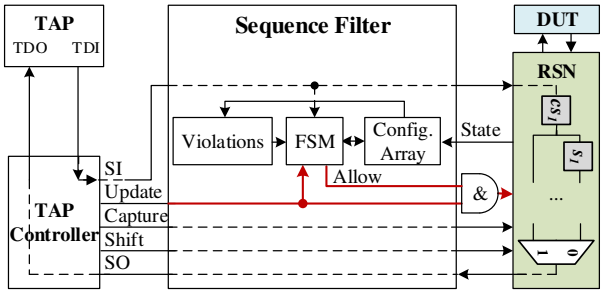

Fig. 3: Details on the filter-based protection

The *Violations* are obtained as a list from the security compliance analysis, considering the explicit and implicit specifications, and stored as conditions into a logic block "Violations". A *Configuration Array* stores the RSN configuration. A *Finite State Machine* (FSM) captures the configuration bits from the input stream at the SI port and allows to keep the configuration array for the current specification updated. It allows to consider complex access requirements for multiple user groups. The filter-based protection does not require any modifications of the RSN structure, even if the security specification changes, e.g. if the trustworthiness of a third-party IP or the information confidentiality have changed.

C. Resynthesis: Structural Changes

Resynthesis allows to resolve all violations, without sacrificing the instruments' accessibility. In [21] the security preserving resynthesis of the RSN is modeled as a minimum cut problem in a multicommodity flow handled by an efficient divide-and-conquer-based heuristic. A minimized number of edges is removed from the G^{RSN} to preclude the violating connectivities. The accessibility of the affected vertices is reintroduced sequentially, following specific criteria, such as the minimized access latency or hardware overhead. The compliance of the RSN is validated again using the SCA, and if some violations still exist, the heuristic is repeated until all the violations are resolved.

More details on the security compliance analysis, the sequence filter and the resynthesis are given in [15, 17, 21].

IV. Filter Applicability Analysis (FAA)

The majority of violations can be resolved by using a sequence filter but some scan segments may become inaccessible through the RSN against the specification, as already shown in Fig. 1. This section presents the first Filter Applicability Analysis (FAA), which allows to identify a maximized set of violations, which can be resolved using a sequence filter without affecting the accessibility of other segments. For each primitive s_j, an essential condition $f(s_j, cs_1, ..cs_n)$ defines the assignment to the control signal values $(cs_1, ..cs_n)$, required to put this primitive into an ASP, and is represented in a conjunctive normal form (CNF). The essential condition is computed iteratively, starting from the scan-out *SO*, and considers the control signals required to select the appropriate branches of multiplexers, and to trigger the *select*-signals (see example of Fig. 4).

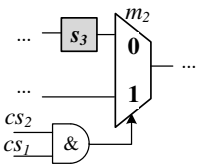

Fig. 4: An active scan path includes s_3, if the appropriate branch of the multiplexer m_2 is selected by setting cs_1 or cs_2 to zero.

A. Security Compliance Violation $viol_{j,k}$

Definition 3: A security compliance violation $viol_{j,k}$ is called *resolvable* by a filter, if for each of the scan segments s_j, s_k there would exist at least a single configurable ASP, which includes this segment.

The applicability of a filter to resolve such a violation can be checked by the following flow:

1) Compute the essential conditions $f(s_j, cs_1, ..cs_n)$, $f(s_k, cs_1, ..cs_n)$ and ensure that each of the segments s_j, s_k is accessible through at least one ASP in the RSN.
2) Verify if data from s_j can be transmitted to s_k using a single scan configuration. A SAT instance is formed to find at least one assignment for the control signal values $cs_1, ...cs_n$, which put both scan segments into an ASP simultaneously:

$$\exists cs_1, .., cs_n : f(s_j, cs_1, ..cs_n) \& f(s_k, cs_1, ..cs_n) := True \quad (1)$$

3) Verify if two ASPs can be configured, such that the first one is used to access s_j only, but should not traverse s_k.

$$\exists cs_1, .., cs_n : (f(s_j, cs_1, ..cs_n) \& \overline{f(s_k, cs_1, ..cs_n)}) := True \quad (2)$$

The second ASP is used to access s_k only.
4) If both paths can be configured, the violation is resolvable by a filter, otherwise it must be resolved structurally.

If it is not possible to assign control values such that Eq.(1) is satisfied but s_j and s_k are functionally connected, the data transfer between these scan segments requires retargeting by using more than one scan configuration. Hence, there are two ASPs, such that the first one traverses only s_j but not s_k, and the second ASP traverses s_k. This means that the violation is resolvable by a filter and a step 3 can be skipped. The same idea is used to verify, if the groups of the scan segments $G_j(u_i), G_k(u_i) \subset S$ are accessible individually.

TABLE I: Access of segments from Fig. 1. Both s_1, s_2 are selected, if $(cs_1, cs_2) := (0, X)$. An assignment to access the s_1 or s_2 individually does not exist, and the violation $viol_{1,2}$ is not resolvable using a filter.

cs_1	cs_2	m_1	m_2	s_1	s_2	s_3	s_4
0	X	0	0	+	+	+	−
1	0	1	0	−	−	+	−
1	1	1	1	−	−	−	+

B. Authorization Violation $viol_j$

Definition 4: An authorization violation $viol_k$ is called *resolvable* by a filter, if after applying a filter, for all the users u_l and for all the required segments $s_{jm} \in A(u_l)$, there exist at least one ASP, which does not traverse the restricted segment $s_k \in G(u_l)$, but traverses the required segment $s_{jm} \in A(u_l)$.

The applicability of a filter for a violation $viol_k$ can be verified using the same logic as in Section IV-A with s_k as a source and all s_{jm} as destinations.

For each segment $\forall s_{jm} \in A(u_l)$, which must remain accessible, an assignment for the control signals $cs_{1m}, ...cs_{nm}$ is searched, which includes s_{jm} into an ASP but does not include any of the restricted segments $s_k \in G(u_i)$. If all such assignments are found, the authorization specification can be fulfilled by a filter without causing any authorization violation.

Example:

$$\exists cs_{1m}, ..cs_{nm} : f(s_{jn}, cs_{1m}, ..cs_{nm}) \& [\cap_{k=1}^{|G(u_i)|} (\overline{f(s_k, cs_{1m}, ..cs_{nm})})] := True \quad (3)$$

Access restriction to cs_1 for u_l (in Fig. 1), makes all the scan segments inaccessible through the RSN. If access to s_1 is required for u_l, the violation is unresolvable by using a filter.

C. Filter Compliance of an RSN

Definition 5: An RSN is called *filter-compliant*, if all of the security compliance violations as well as all of the authorization violations are resolvable using a sequence filter without blocking the access to other segments.

The FAA is applied to all the violations sequentially in order to verify, whether a given RSN is filter-compliant. If the RSN is filter-compliant, the information about the violations can be further used to guide the filter generation. In the other case, the violating connectivities, which are unresolvable using a filter, can be resolved structurally by using resynthesis.

V. General Protection Flow

The hybrid protection scheme (Fig. 5) consists of two steps:

1) *Prepare the RSN*: an initial RSN is transformed into a *filter-compliant* RSN (FC-RSN in Fig. 5) using a minimized number of changes.

2) *Generate the Filter*: a sequence filter is generated for the filter-compliant RSN according to [17] to resolve the remaining violations.

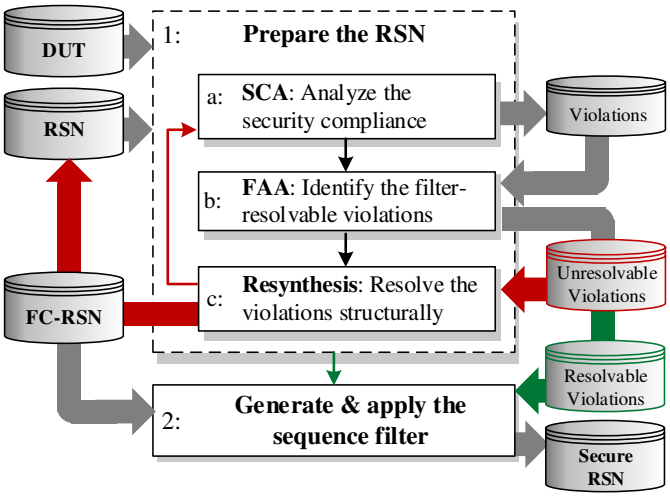

Fig. 5: General flow of the hybrid protection scheme

A secure RSN implementation, which preserves the specified accessibility of the segments, consists of a filter-compliant RSN and the generated sequence filter. This implementation combines the flexibility and online usability of the filter with the generality of the resynthesis, and allows to resolve any security compliance violation. The remainder of the section provides details on the individual steps.

A. Prepare the RSN

Following steps are required to resynthesize the RSN:

1) **SCA:** The reachability properties of the DUT and the RSN are computed and the security compliance of the RSN with the DUT is verified. If the RSN is not compliant, the list of authorization and security compliance violations is generated.
2) **FAA:** The list of the violations is analyzed using the Filter Applicability Analysis presented in Section IV, and the violations are divided into two subsets. The first subset includes the violations, which can only be resolved by applying the structural changes, the second subset - the violations resolvable by using a sequence filter.
3) **Resynthesis:** The resynthesis approach, as presented in [21], is adjusted to obtain a filter-compliant RSN with a minimized number of changes and is only applied to the subset of violations, which are unresolvable by a filter.
   - A minimized number of edges is removed from G^{RSN} to cut the violating connectivities, which cannot be resolved using filters.
   - The accessibility of the affected vertices is reintroduced sequentially.
   - For each vertex s_j, having no successors, a prioritized list of possible successors $PS(s_j)$ is formed. $PS(s_j)$ contains all such vertices s_k that a newly introduced connectivity between s_j and s_k would be resolvable by a filter, even if the specification changes.
   - The highest priority is given to such resolvable connectivities, which do not cause a security compliance violation in the current security specification.
   - If multiple vertices in $PS(s_j)$ have the highest priority, additional optimization criteria can be specified by a test engineer, e.g. a minimized access latency of safety-critical instruments or hardware overhead. For any vertex s_j, $PS(s_j)$ is not empty, since it includes the auxiliary Scan-Out vertex.
   - The same idea is applied to reintroduce the accessibility of the vertices with no predecessors.

B. Validate the Filter Applicability

The RSN generated in step 1, is analyzed again to check, whether all the violations are already resolved and to generate the updated list of violations otherwise. The updated list is analyzed for the filter applicability and the subsets of violations are recomputed, since the structural changes can affect the filter applicability. The RSN is modified, until all the violations from the list are resolvable by the filter. This incurs minor number of structural changes, and offers flexibility for the future, if the specification will change. The presented scheme is guaranteed to converge to a filter-compliant RSN and the remaining violations are resolved using a filter. In the worst case, a parallel RSN structure, where all the instruments are accessed using the scan segments, located in the different branches of a scan multiplexer, is obtained. Experiments in Section VI show that the computation converges with just a minor number of structural changes.

C. Generate and Apply the Sequence Filter

The list of violations, which have not been structurally resolved, and the modified filter-compliant RSN are used for the filter [17] generation. The filter handles the remaining violations online and concurrently and prohibits any violating access.

VI. Experimental Results

The experiments have been conducted on Intel(R) Xeon(R) W-2125 CPU at 4.00GHz with 132 GB of main memory. A subset of benchmarks from the ITC'16 [27] benchmark set and the largest benchmarks from the industrial DATE'2019 benchmark set [20] are used as the initial RSNs. For a detailed information about the properties of the RSN benchmarks refer to [21]. The connectivities in the DUT are represented by the benchmarks from the ISCAS'89 benchmark set [28]. Each flip-flop represents a register of an instrument and one segment in the RSN is used to access one instrument register.

The experimental results for the whole flow from Fig. 5 are shown in Table II. Two additional user groups have been considered besides the complete access for the manufacturer: one for the system integrators, and one for maintenance and user. The experiments deal with the general case with conflicting rights. For each benchmark, the number of the identified violations $\#viol$ is presented in Column 2. Column 3 shows the number of iterations of the "Prepare the RSN" step of the general flow, which have been required to synthesize a filter-compliant RSN.

TABLE II: A Hybrid Protection Scheme

Initial Design		Overhead				Comparison		
(1) Benchmark	(2) $\#viol.$	(3) $\#it.$	(4) $\%\triangle\ HW_{filter}$	(5) t_{struct}[m:s]	(6) t_{filter}[m:s]	(7) % resolvable by filters only	(8) #RSN changes hybrid	(9) #RSN changes struct [21]
q12710	501	3	0.98	00:05	00:51	84.0	38	63
a586710	1.820	2	0.78	02:10	01:46	67.3	47	186
p34392	502	3	4.17	00:30	01:48	79.2	48	71
t512505	924	2	1.65	02:54	00:59	94.6	36	174
p22810	28,941	2	7.21	06:40	10:25	93.1	499	1,922
p93791	18,610	2	5.50	08:34	13:51	94.2	463	1,891
MBIST_2_20_20	8,519	1	10.05	00:58	01:36	83.4	1,408	2,247
MBIST_5_20_20	1,020	0	6.42	01:20	02:16	100	0	125
MBIST_5_100_20	2,559	1	23.79	14:33	10:23	90.0	241	415
MBIST_5_100_100	9,828	1	3.44	18:46	19:54	99.2	78	1,020
MBIST_20_20_20	3,779	1	3.41	14:01	09:27	89.8	293	335
MBIST_55_20_5	59,055	1	8.90	10:25	11:36	85.1	6,857	7,658
MBIST_100_20_5	22,084	1	10.54	05:48	07:23	98.3	354	1,563
MBIST_100_100_5	160,687	1	8.63	12:54	25:10	99.8	195	10,376

The area overhead of the generated filter $\%\triangle\ HW_{filter}$ is shown in Column 4 and is defined as a ratio between the filter size in terms of scan cells compared to the total RSN size in terms of scan cells. The runtimes to obtain a filter-compliant RSN and to generate the filter are given in Columns 5 and 6 respectively.

Definition 6: The *filter applicability fraction* is a fraction of the number of violations, which are resolvable by a filter, $\#viol_{filter}$ compared to the total number of violations $\#viol$:

$$\%resolvable\ by\ filters = \frac{\#viol_{filter}}{\#viol} * 100\% \qquad (4)$$

The filter applicability fraction is shown in Column 7 for all the benchmarks. Hardware overhead is measured as the number of modified direct connectivities in terms of the RSN graph edges. This number is provided in Column 8 for the presented hybrid method. The same metric is provided for the pure resynthesis method of [21] in Column 9.

A. Hybrid Protection Scheme Results

For all the benchmarks, all the violations (Column 2) have been resolved. After a few of iterations of the "Prepare the RSN" step (Column 3), a filter-compliant RSN is obtained. Just a minor number of structural changes (Column 8) was required for all the considered benchmarks to transform the RSN into a filter-compliant RSN. The remaining violations are resolved using a filter. The area overhead (Column 4) of the generated filter is acceptable for all the RSNs. In order to preserve an acceptable complexity of a secure filter for the hierarchically organized DATE benchmarks, the filter is also constructed hierarchically out of multiple FSMs. Each FSM corresponds to a sub-RSN, which is used to access a part of the instruments, such as the memory BIST registers. The coordination between the individual FSMs can be implemented as in [29] and lays out of the scope of the paper. As the size and the complexity of the benchmark increases, the relative area overhead decreases and becomes negligible for the largest RSNs. The runtime (Columns 5 and 6) is acceptable even for the largest benchmarks.

B. Comparison to the State of the Art

1) Filters: In contrast to the pure filter-based approach, as in [17] all the violations have been resolved without sacrificing the instruments' accessibility. The filter applicability fraction (Column 7) shows that only for one benchmark, applying a pure filter-based approach to resolve the violations would not make any required instrument inaccessible via the RSN.

2) Resynthesis: Compared to the pure structural solution [21] (Column 9), less hardware overhead is required (Column 8). Since the majority of violations is resolved by a filter in a hybrid scheme, the number of required structural changes to the RSN is dramatically decreased. Moreover, the hybrid approach is able to specify different access rights for various user groups, which is not possible by using resynthesis alone.

VII. Conclusion

This paper presents an automated hybrid protection scheme for RSNs, which combines the benefits of the functional and structural approaches, and overcomes their limitations. A Filter Applicability Analysis identifies violations not resolvable by any sequence filter. A minimized number of structural transformations is applied to modify the "filter-unresolvable" connectivities inside the RSN in a way that all the remaining violations can be resolved using a filter. Finally, a flexible filter is generated to resolve the remaining violations.

The resulting protection scheme considers complex security-preserving access scenarios for multiple user groups with specific permissions, and does not require additional changes to the RSN structure, even if the security specification changes, while preserving the accessibility of all required instruments through the RSN. Only a minor part of the violations ($\leq 11\%$) is resolved structurally and the hardware costs are dramatically reduced. The experimental results show a good scalability even for the largest benchmarks.

Acknowledgments

This work was supported by the German Research Foundation (DFG) under grant WU 245/17-2 (ACCESS) and partially supported by Advantest as part of the Graduate School "Intelligent Methods for Test and Reliability" (GS-IMTR) at the University of Stuttgart.

BIBLIOGRAPHY

[1] "IEEE Standard for Access and Control of Instrumentation Embedded within a Semiconductor Device," *IEEE Std 1687-2014*, pp. 1–283, Dec. 2014.

[2] "IEEE Standard for Test Access Port and Boundary-Scan Architecture," *IEEE Std 1149.1-2013 (Revision of IEEE Std 1149.1-2001)*, pp. 1–444, May 2013.

[3] K. Shibin, S. Devadze, A. Jutman, M. Grabmann, and R. Pricken, "Health Management for Self-Aware SoCs Based on IEEE 1687 Infrastructure," *IEEE Design Test*, vol. 34, no. 6, pp. 27–35, Dec. 2017.

[4] M. A. Kochte, M. Sauer, L. R. Gomez, P. Raiola, B. Becker, and H. Wunderlich, "Specification and Verification of Security in Reconfigurable Scan Networks," in *Proc. IEEE European Test Symp. (ETS)*, May 2017, pp. 1–6.

[5] J. Dworak and A. Crouch, "A Call to Action: Securing IEEE 1687 and the Need for an IEEE Test Security Standard," in *Proc. IEEE VLSI Test Symp. (VTS)*, Apr. 2015, pp. 1–4.

[6] J. Lee, M. Tehranipoor, C. Patel, and J. Plusquellic, "Securing Designs against Scan-Based Side-Channel Attacks," *IEEE Trans. on Dependable and Secure Computing (TDSC)*, vol. 4, no. 4, pp. 325–336, Nov. 2007.

[7] X. Wang, D. Zhang, M. He, D. Su, and M. Tehranipoor, "Secure Scan and Test Using Obfuscation Throughout Supply Chain," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 37, no. 9, pp. 1867–1880, Sept. 2018.

[8] R. Elnaggar, R. Karri, and K. Chakrabarty, "Securing IJTAG against data-integrity attacks," in *Proc. IEEE VLSI Test Symp. (VTS)*, Apr. 2018, pp. 1–6.

[9] S. Kan and J. Dworak, "IJTAG Integrity Checking with Chained Hashing," in *Proc. IEEE Int'l Test Conf. (ITC)*, Oct. 2018, pp. 1–10.

[10] X. Ren, R. D. S. Blanton, and V. G. Tavares, "Detection of IJTAG attacks using LDPC-based feature reduction and machine learning," in *Proc. IEEE European Test Symp. (ETS)*, May 2018, pp. 1–6.

[11] M. A. Kochte, R. Baranowski, and H.-J. Wunderlich, "Trustworthy Reconfigurable Access to On-Chip Infrastructure," in *Proc. IEEE Int'l Test Conf. in Asia (ITC-Asia)*, Sep. 2017.

[12] J. Dworak, A. Crouch, J. Potter, A. Zygmontowicz, and M. Thornton, "Don't forget to lock your SIB: hiding instruments using P1687," in *Proc. IEEE Int'l Test Conf. (ITC)*, Sept. 2013, pp. 1–10.

[13] A. Zygmontowicz, J. Dworak, A. Crouch, and J. Potter, "Making It Harder to Unlock an LSIB: Honeytraps and Misdirection in a P1687 Network," in *Proc. Conf. Design, Automation & Test in Europe (DATE)*, Mar. 2014, pp. 195:1–195:6.

[14] R. Baranowski, M. A. Kochte, and H.-J. Wunderlich, "Fine-Grained Access Management in Reconfigurable Scan Networks," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 34, no. 6, pp. 937–946, Jun. 2015.

[15] N. Lylina, A. Atteya, P. Raiola, M. Sauer, B. Becker, and H.-J. Wunderlich, "Security Compliance Analysis of Reconfigurable Scan Networks," in *Proc. IEEE Int'l Test Conf. (ITC)*, Nov. 2019, pp. 1–9.

[16] R. Baranowski, M. A. Kochte, and H.-J. Wunderlich, "Access Port Protection for Reconfigurable Scan Networks," *Journal of Electronic Testing: Theory and Applic. (JETTA)*, vol. 30, no. 6, pp. 711–723, 2014.

[17] A. Atteya, M. A. Kochte, M. Sauer, P. Raiola, B. Becker, and H.-J. Wunderlich, "Online Prevention of Security Violations in Reconfigurable Scan Networks," in *Proc. IEEE European Test Symp. (ETS)*, May 2018, pp. 1–6.

[18] A. L. Crouch, B. G. Van Treuren, and J. Rearick, "P1687.1: Accessing Embedded 1687 Instruments using Alternate Device Interfaces other than JTAG," in *Proc. IEEE European Test Symp. (ETS)*, May 2020, pp. 1–6.

[19] P. Raiola, M. A. Kochte, A. Atteya, L. R. Gomez, H.-J. Wunderlich, B. Becker, and M. Sauer, "Detecting and Resolving Security Violations in Reconfigurable Scan Networks," in *Proc. IEEE Int'l Symp. on On-Line Testing And Robust System Design (IOLTS)*, Jul. 2018, pp. 91–96.

[20] P. Raiola, B. Thiemann, J. Burchard, A. Atteya, N. Lylina, H.-J. Wunderlich, B. Becker, and M. Sauer, "On Secure Data Flow in Reconfigurable Scan Networks," in *Proc. Conf. on Design, Automation Test in Europe (DATE)*, Mar. 2019, pp. 1–6.

[21] N. Lylina, A. Atteya, C.-H. Wang, and H.-J. Wunderlich, "Security Preserving Integration and Resynthesis of Reconfigurable Scan Networks," in *Proc. IEEE Int'l Test Conf. (ITC)*, Nov. 2020, pp. 1–10.

[22] R. Baranowski, M. A. Kochte, and H.-J. Wunderlich, "Reconfigurable Scan Networks: Modeling, Verification, and Optimal Pattern Generation," *ACM Trans. Design Automation of Electronic Systems (TODAES)*, vol. 20, no. 2, pp. 30:1–30:27, 2015.

[23] Z. Hanna and V. M. Purri. (2013, Apr.) Verifying Security Aspects of SoC Designs with Jasper App. [Online]. Available: https://www.edn.com/

[24] K. Nakamura, K. Takagi, S. Kimura, and K. Watanabe, "Waiting false path analysis of sequential logic circuits for performance optimization," in *Proc. IEEE/ACM Int'l Conf. on Computer-Aided Design. Digest of Technical Papers (ICCAD)*, Nov 1998, pp. 392–395.

[25] A. Ardeshiricham, W. Hu, J. Marxen, and R. Kastner, "Register transfer level information flow tracking for provably secure hardware design," in *Proc. Conf. on Design, Automation Test in Europe (DATE)*, Mar. 2017, pp. 1691–1696.

[26] M. Soeken, P. Raiola, B. Sterin, B. Becker, G. De Micheli, and M. Sauer, *Proc. 12th Int'l Haifa Verification Conference (HVC)*. Springer, 2016, ch. SAT-Based Combinational and Sequential Dependency Computation, pp. 1–17.

[27] A. Tsertov, A. Jutman, S. Devadze, M. S. Reorda, E. Larsson, F. G. Zadegan, R. Cantoro, M. Montazeri, and R. Krenz-Baath, "A suite of IEEE 1687 benchmark networks," in *Proc. IEEE Int'l Test Conf. (ITC)*, Nov. 2016, pp. 1–10.

[28] F. Brglez, D. Bryan, and K. Kozminski, "Combinational profiles of sequential benchmark circuits," in *Proc. Int'l Symp. on Circuits and Systems (ISCAS)*, May 1989, pp. 1929–1934 vol.3.

[29] D. Harel, "Statecharts: a visual formalism for complex systems," *Science of Computer Programming*, vol. 8, no. 3, pp. 231 – 274, 1987.