Securing Access to Reconfigurable Scan Networks

Baranowski, Rafal; Kochte, Michael A.; Wunderlich, Hans-Joachim

Proceedings of the 22nd IEEE Asian Test Symposium (ATS'13) Yilan, Taiwan, 18-21 November 2013

doi: http://dx.doi.org/10.1109/ATS.2013.61

Abstract: The accessibility of on-chip embedded infrastructure for test, reconfiguration, and debug poses a serious safety and security problem. Special care is required in the design and development of scan architectures based on IEEE Std. 1149.1 (JTAG), IEEE Std. 1500, and especially reconfigurable scan networks, as allowed by the upcoming IEEE P1687 (IJTAG). Traditionally, the scan infrastructure is secured after manufacturing test using fuses that disable the test access port (TAP) completely or partially. The fuse-based approach is efficient if some scan chains or instructions of the TAP controller are to be permanently blocked. However, this approach becomes costly if fine-grained access management is required, and it faces scalability issues in reconfigurable scan networks. In this paper, we propose a scalable solution for multi-level access management in reconfigurable scan networks. The access to protected registers is restricted locally at TAP-level by a sequence filter which allows only a precomputed set of scan-in access sequences. Our approach does not require any modification of the scan architecture and causes no access time penalty. Experimental results for complex reconfigurable scan networks show that the area overhead depends primarily on the number of allowed accesses, and is marginal even if this number exceeds the count of network's registers.

Preprint

Securing Access to Reconfigurable Scan Networks

Rafal Baranowski, Michael A. Kochte, Hans-Joachim Wunderlich
ITI, University of Stuttgart, Pfaffenwaldring 47, D-70569, Stuttgart, Germany
Email: {baranowski, kochte}@iti.uni-stuttgart.de, wu@informatik.uni-stuttgart.de

***Abstract*—The accessibility of on-chip embedded infrastructure for test, reconfiguration, and debug poses a serious safety and security problem. Special care is required in the design and development of scan architectures based on IEEE Std. 1149.1 (JTAG), IEEE Std. 1500, and especially reconfigurable scan networks, as allowed by the upcoming IEEE P1687 (IJTAG).**

Traditionally, the scan infrastructure is secured after manufacturing test using fuses that disable the test access port (TAP) completely or partially. The fuse-based approach is efficient if some scan chains or instructions of the TAP controller are to be permanently blocked. However, this approach becomes costly if fine-grained access management is required, and it faces scalability issues in reconfigurable scan networks.

In this paper, we propose a scalable solution for multi-level access management in reconfigurable scan networks. The access to protected registers is restricted locally at TAP-level by a *sequence filter* which allows only a precomputed set of scan-in access sequences. Our approach does not require any modification of the scan architecture and causes no access time penalty. Experimental results for complex reconfigurable scan networks show that the area overhead depends primarily on the number of allowed accesses, and is marginal even if this number exceeds the count of network's registers.

***Index Terms*—Debug and diagnosis, reconfigurable scan network, IJTAG, IEEE P1687, secure DFT, hardware security**

I. Introduction

A large fraction of integrated systems is devoted to embedded instrumentation that facilitates test, diagnosis, or post-silicon validation. Such on-chip instrumentation is also used during operation in the field for power-up initialization, reconfiguration, monitoring, error management, fault tolerance or repair [1], [2], [3].

Reconfigurable Scan Networks (RSNs) emerge as a scalable and flexible option for efficient and cost-effective access to embedded instrumentation. The ongoing effort IEEE Std. P1687 (or IJTAG) aims to standardize the design and access to such scan networks, extending the widely adopted IEEE Std. 1149.1 (JTAG). This allows flexible architectures with distributed and hierarchical configuration for efficient instrumentation access [1].

The accessibility of on-chip infrastructure contradicts with security and safety requirements for chip internals [4]. An attacker may exploit the scan infrastructure to gain access to protected data (secret key or IP), alter the system state to perform illegal operations [5], or perform side-channel attacks, e.g. on cryptographic cores [6]. Unintended activation of the test or debug infrastructure, e.g. due to a hardware fault or a soft error, may also lead to the violation of safety properties.

Different levels of infrastructure accessibility are required e.g. during test, bring-up and post-silicon validation, as well as during regular operation for system maintenance and reliability improvement. For example in automotive applications, full access is required during manufacturing and assembly test, while only limited access is allowed during operation and maintenance in a workshop to prevent tampering.

Logical security of scan infrastructure based on authentication and authorization increases the protection level but cannot completely prohibit physical access. The protection methods proposed in literature include access authorization [7], [8], [9], shift data encryption [9], and scan chain obfuscation [7], [10]. To protect individual structures from unauthorized access, encryption circuitry has to be distributed over the chip [11], with high hardware cost.

The goal of these approaches is to assure that only users who know a shared secret (e.g. encryption key, or obfuscation principle) can access the scan infrastructure. If the shared secret is known to the attacker, full access becomes possible which is unacceptable in safety critical applications.

To guarantee inaccessibility of protected registers, the physical interface or parts of scan infrastructure can be made unusable, e.g. using on-chip fuses [12]. This prevents any access to protected structures at the cost of reduced flexibility and accessibility: By blowing an on-chip fuse, some instructions of the TAP controller or chosen scan chains can be permanently disabled [13]. The traditional fuse-based protection requires thorough consideration early in the design process and is difficult or even impossible in core-based design flows. For fine-grained access management, as well as in hierarchical RSNs, a high number of fuses may need to be integrated and distributed over the entire network, causing significant hardware overhead of fuses and their control circuitry [12].

We propose a scalable approach that offers flexible multi-level access management for RSNs. We extend the TAP with a *sequence filter* (Fig. 1) that only allows access to a subset of (unprotected) scan registers and denies the access to protected registers. If required, the filter can be enabled by a single fuse, e.g. after manufacturing test. Our approach can be employed in core-based design flows, allowing fine-grained access management. It is directly applicable to complex RSNs compliant with IEEE P1687 and requires just a high-level network description in Instrument Connectivity Language (ICL [1]), or a precomputed set of scan-in data for allowed accesses.

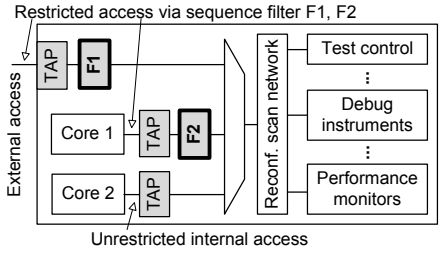

Fig. 1. Example of access restriction based on *sequence filters*

Compared with fuse-based protection, our approach offers a similar protection level, but does not require any dedicated scan architecture nor does it change the existing internal access mechanism. We allow fine-grained access management

at TAP-level: For instance in Fig. 1, the filter F1 can be used to block the external accessibility of scan chains, while full accessibility is preserved for debugging purposes via the internal TAP of "Core 2". A sequence filter can also be used to allow individual (exclusive) access to a set of instruments, and still block simultaneous (concurrent) access to them, e.g. to prevent that sensitive data is shifted through exposed or untrusted instruments. In addition, our approach can be combined with authentication mechanisms to provide logical security without the need to redesign the scan network.

In the next section, we present the problem statement, provide an overview over the proposed approach, and give a short introduction to RSNs. In Section III we formally define restricted accesses and present an algorithm for construction of sequence filters. The area overhead is evaluated in Section IV.

II. Overview

Reconfigurable scan networks are usually accessed through a JTAG-compliant Test Access Port (TAP). An RSN can be viewed as a reconfigurable Data Register (DR in IEEE Std. 1149.1/JTAG) with *variable length*. The logic state of the RSN determines which scan registers in the network are currently accessible. The RSN state is changed by rewriting the content of accessible registers.

Our goal is to restrict the accessibility of an RSN at TAP-level, without changing the RSN architecture. We aim to provide the access to a subset of specified (target) scan registers (e.g. temperature monitors) and restrict the access to protected registers (e.g. internal test structures or scan chains).

A. Problem Formulation

Given is a reconfigurable scan network with a set of *protected* scan registers that must not be accessible through a physical interface (e.g. a TAP). We aim to restrict the interface so that it is impossible to access (read from or write to) the protected registers through this interface, while the access to other (target) registers remains possible.

We achieve this goal with a *sequence filter* that blocks forbidden accesses at the TAP. The sequence filter is placed between the TAP and the scan network, as shown in Fig. 2. It observes the scan-in data and decides whether the access to the RSN is allowed or forbidden. If the scan-in data does not provide access to any protected scan registers, the filter does not interfere in the scan operation. Otherwise, the filter prevents all registers from latching the shift-in data by inhibiting the *update* control signal, as explained in Section II-B.

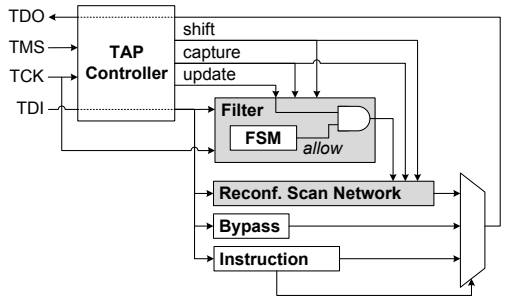

Fig. 2. Access restriction at the Test Access Port (TAP)

The sequence filter requires only a minor extension of the TAP, without modification of the internal or external TAP interface. In particular, the proposed method requires neither modifications of the RSN architecture, nor addition of any global control signal. This makes our approach well-suited for core-based design with hard IP cores, or even for 3D integration of fixed Known Good Dies (KGD).

An overview of the proposed method is presented in Fig. 3. The scan-in data for restricted access is generated with the approach from [14], [15]. The sequences form an input to the filter construction algorithm, as discussed in Section III-B.

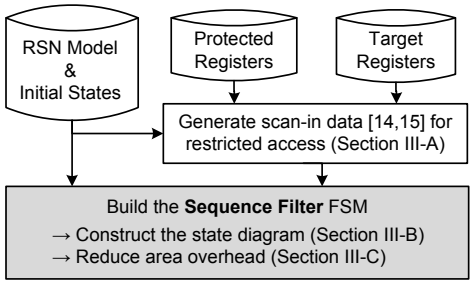

Fig. 3. Overview of the proposed method

B. Reconfigurable Scan Networks

An example of a simple RSN is given in Fig. 4. The one-bit scan registers S1 and S3 control the access to two multi-bit scan registers S2 and S4, respectively. The scan-in data is shifted through registers S2 and S4 only if the previous access assured that $S1 = S3 = 1$. The path through which the scan-in data is shifted is also referred to as the *active scan path*.

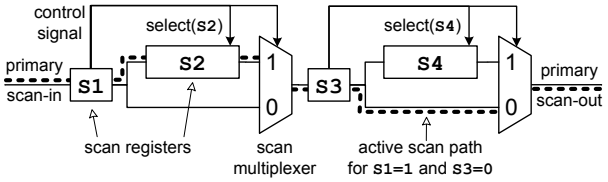

Fig. 4. Example of a reconfigurable scan network and its terminology

The basic access to the RSN is an atomic (inseparable) operation. It consists of three phases that are managed by the JTAG TAP controller: Capture, Shift, and Update (CSU). During *capture*, the scan registers on the active scan path may latch new data. This data is shifted out during the *shift* phase, while new scan data is shifted in. Finally, during the *update* phase, the shifted-in data is latched in the *shadow latches* of the scan registers on the active scan path, which, in turn, may change the flow of the active scan path. The update phase is *blocked by the sequence filter* (cf. Fig. 2) if the shifted-in data is not allowed. This assures that the active scan path is stable and the protected registers are not exposed.

The state of all scan registers in the RSN is referred to as *scan configuration*. A read or write access to a scan register in the network requires that the accessed register is part of an active scan path in the current scan configuration (cf. Fig. 4). A *scan access* is a sequence of CSU operations required to reconfigure the RSN and access the target register. Multiple registers may be read or written *concurrently* in one access.

Our method poses just two requirements on the RSN: In the initial (reset) scan configuration, no protected scan register may belong to the active scan path, and there must exist a way to bypass protected scan registers while accessing other

registers. If the access to a certain target register requires that a given protected scan register be modified or exposed, the protected register needs to be extended with a configurable bypass, e.g. a Segment Insertion Bit (SIB), as in [1].

III. Access Restriction at TAP-level

A. Restricted Access

Definition 3.1: Given a set of protected registers, a set of target registers, and a set of initial scan configurations I of an RSN, we define the *restricted access* as a scan access such that:

- The access has the same effect on the scan network, i.e. the target registers are properly accessed, for all initial scan configurations in I.
- During the access, no protected scan registers belong to the active scan path (the shift data does not pass through any protected scan register).
- After the access, the final scan configuration of the scan network belongs to I.

From Definition 3.1 it follows that any concatenation of restricted accesses is also a restricted access, as the final scan configuration after the restricted access belongs I.

For example, consider the RSN from Fig. 4. Let us assume that I is defined as the set of all scan configurations in which $\texttt{S1} = \texttt{S3} = 0$. We allow the access to register S2 and assume that S4 is protected. According to Definition 3.1, a restricted access to S2 must guarantee that:

- S2 is accessed for any initial scan configuration satisfying $\texttt{S1} = \texttt{S3} = 0$ (regardless of the content of S2 and S4).
- S4 is never part of the active scan path.
- After the access, the initial scan configuration is restored, i.e. $\texttt{S1} = \texttt{S3} = 0$.

A possible restricted access to register S2 consists of two CSU operations with the following scan-in data (leftmost bit is shifted-in first): *01* and *0X0*, where *X* stands for the target value of S2. The first CSU operation puts register S2 on the active scan path. In the second CSU operation, S2 is accessed, while the initial state of S1 is restored. The protected register S4 is bypassed, and the final scan configuration satisfies $\texttt{S1} = \texttt{S3} = 0$.

We map the access generation to a Boolean satisfiability (SAT) problem instance, and generate restricted accesses with minimal access time using a pseudo-Boolean SAT solver. The SAT instance includes Boolean constraints which ensure that:

- Protected scan registers never belong to the active scan path, i.e. their content is never altered nor exposed.
- After the access, the initial scan configuration is restored.

B. Sequence Filter Construction

A sequence filter consists of a finite state machine (FSM) with four inputs from the TAP controller (test data input; shift, capture, and update control signals) and a single output *allow* which controls the update operation in the RSN (cf. Fig. 2). The FSM tracks the scan operations at the TAP: As long as the sequence of scan operations matches a restricted access, the *allow* signal is active and the access is applied to the RSN without any delay. Otherwise, the FSM enters a deadlock state and signal *allow* is deactivated. This makes the access to protected scan registers impossible via the TAP.

The FSM's state diagram is constructed using the algorithm given in Procedure 1. The input to the procedure (`sequenceSet`) is a set of sequences representing the allowed restricted accesses, i.e. strings composed of the following scan operations: shift of value 0, shift of value 1, shift of an unconstrained (don't care) value, capture, and update. We denote these scan operations by *0*, *1*, *X*, *C*, and *U*, respectively. For instance, a restricted access consisting of two CSU operations with scan-in data *01* and *0X0* is represented by the following sequence: *C01UC0X0U*. Note that a single sequence represents 2^k restricted accesses, where k is the number of unconstrained (*X*) data bits in the sequence.

Procedure 1 Sequence filter construction

Input: `sequenceSet`
Output: state diagram

```
 1: create initialState, deadlockState
 2: annotate initialState with all sequences from sequenceSet
 3: currentStateSet ← {initialState}
 4: while currentStateSet ≠ ∅ do
 5:   for all state ∈ currentStateSet do
 6:     for all sequence ∈ annotations of state do
 7:       transition ← current scan operation in sequence
 8:       if transition = U and sequence is finished then
 9:         add transition from state to initialState
10:       else
11:         create newState and annotate it with sequence
12:         add transition from state to newState
13:         add newState to nextStateSet
14:       end if
15:     end for
16:   end for
17:   replace overlapping transitions from states in currentStateSet
18:   add escape transitions from currentStateSet to deadlockState
19:   merge states in nextStateSet
20:   currentStateSet ← nextStateSet
21:   for all sequence ∈ sequenceSet do
22:     move to next operation in sequence
23:   end for
24: end while
```

Each state in the state diagram is annotated with the sequences that put the FSM into this state. State transitions depend on the current scan operation: The condition of a transition is either a specific scan operation (i.e. an element from the set $\{0, 1, X, C, U\}$), or a disjunction of scan operations (e.g. *C* or *U*, denoted by *C,U*). All states are stable as long as no scan operation takes place.

The diagram construction algorithm starts with the creation of an initial (reset) state (`initialState` in Procedure 1). The remaining part of the algorithm is a stepwise procedure. Each step adds another level of states to the state diagram, based on the scan operations in the provided sequences. In the first step, we deal with the first scan operation of each sequence (i.e., the capture operations). In the n-th step, we check the n-th operation of each sequence. The current scan operation in each sequence is assigned a new successor state (`newState`) with an incoming transition from the respective state in `currentStateSet`. The procedure terminates when all sequences are completely processed.

In each step, after the successor states are found, overlapping shift transitions of each current state are replaced: If a state has both an outbound *X* transition and an outbound *0* (*1*) transition, the *X* transition is replaced with a *1* (*0*) transition, and the annotations of both successors are updated accordingly. An example is given in Fig. 5.

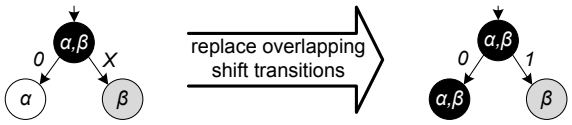

Fig. 5. Example of a state diagram before and after replacement of overlapping shift transitions. Annotations α and β denote two sequences.

At this stage, if all scan operations are allowed in a state, this state has either 3 or 4 outbound transitions, i.e. either $\{C, U, 0, 1\}$, or $\{C, U, X\}$. If some operations are *forbidden* (not allowed by any of the provided sequences), the sequence filter must detect them and prevent any further reconfiguration of the network. To this end, we create an additional state – `deadlockState`. For each state with an incomplete set of outbound transitions, we add an *escape transition* that points to the `deadlockState` and is taken upon detection of a forbidden operation. Once a forbidden operation is encountered, the filter is stuck in the `deadlockState`. In this state, the update operation is inhibited, hence any further reconfiguration is disabled until the filter and the scan network are reset. Section III-D provides an example.

The filter's state must be synchronized with the scan configuration of the protected RSN: The reset signal must reliably put both the RSN and the sequence filter to their initial states. If the RSN is accessed through another TAP (e.g. via an internal interface), the sequence filter is put to its deadlock state, and the infrastructure is reset before the next restricted access. This assures that no forbidden access can take place when the sequence filter is not synchronized.

To guarantee security in presence of soft errors and hardware defects in the filter FSM, the FSM can be designed fail-safe [16]: In presence of faults, the FSM's output *allow* must be either correct or inactive (0).

C. State Diagram Reduction

To reduce the size of the state diagram, in each step we remove redundancies by merging states among the newly created successors in `nextStateSet`. Two successor states can be merged if they fulfill one of the following conditions:

- The two states have identical annotations.
- The states have the same type of an incoming transition, and their predecessors have the same annotations.

The state that results from merging two states receives all annotations of its constituent states.

The resulting state diagram often includes long sequences of consecutive shift operations with constant or unconstrained (*X*) values, as in Fig. 6a. Typically, long sequences of *X* operations represent unconstrained data for target registers. We merge such sequences into a single state, and use a counter to keep track of their length. Just a single counter is required regardless of how many sequences are merged. Our experimental results show that sequence merging reduces the area overhead by an order of magnitude.

Fig. 6 presents a merging example for the sequence *XXX1*. The states b, c, d in Fig. 6a are merged into a single state m in Fig. 6b. During the transition to the merged state m, the counter `cnt` is set to 2. Unless a *C* or *U* operation occurs, the counter is decreased by any shift operation (*0*, *1*, or *X*). The state m is left when either the counter is zero, or a forbidden operation (*C* or *U*) is detected.

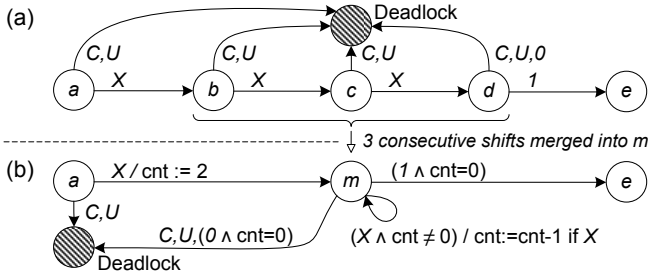

Fig. 6. Example for sequence merging with (a) a state diagram and (b) its merged equivalent

A sequence filter can be optimized to allow repeated accesses to a set of target scan registers with little or no hardware overhead. This is crucial to apply many patterns to the same registers with no access time penalty. To enable such repeated accesses, the state diagram is extended with a loop transition. This is explained at an example in the next section.

D. Example

In the following, we present an example of a sequence filter for the RSN from Fig. 4. We make the same assumptions as in the example from Section III-A: Register `S4` is protected, and the set of initial scan configurations is defined by $\texttt{S1} = \texttt{S3} = 0$. We allow two restricted accesses to register `S2`, characterized by the following sequences α and β:

- α: *C01UC0X0U* (as in Section III-A),
- β: *C01UC0X1UC0X0U*, which accesses `S2` twice.

Fig. 7 presents the resulting state diagram of the filter. The annotations of states (α and β) are denoted inside the state symbols. For the sake of clarity, the escape transitions to the deadlock state are shown only for the first three states.

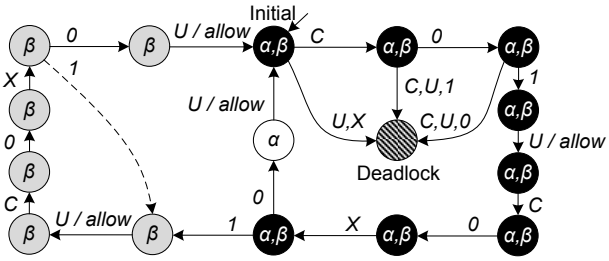

Fig. 7. The state diagram of a filter allowing two sequences, α: *C01UC0X0U*, and β: *C01UC0X1UC0X0U*

The filter can be extended with a single loop transition to allow repeated accesses to `S2` without the need to reconfigure `S1` in each access. This transition is dashed in Fig. 7.

IV. Evaluation

To evaluate the cost of the proposed protection method, we build the sequence filters for complex RSNs based on ITC'02 benchmarks [17]. We evaluate the area overhead of the sequence filters for individual and concurrent restricted accesses to random scan registers.

A. Benchmark Circuits

Our approach is evaluated on two RSN architectures from [14]: hierarchical structures implemented with multiplexers (MUX), and Segment Insertion Bits (SIB).

The *SIB-based* scan architecture implements hierarchical scan bypasses with SIBs, which consist of a 1-bit configuration register and a scan multiplexer that either bypasses or connects the lower-level scan register (or a scan network hierarchy) to the higher-level scan chain, depending on the content of the configuration register [18].

The *MUX-based* architecture supports two modes: configuration access and data access. Configuration access allows to reconfigure the scan chain by attaching or detaching internal scan registers or sub-modules. More details are given in [14].

Due to space limitations, we only consider benchmark circuits with more than 10 000 scan flip-flops (bits), as listed in Table I. For MUX-based architectures, the number of multiplexers is given in the second column, the total number of scan registers (including configuration registers) in the third column, and the total number of scan flip-flops (bits) in the fourth column. The characteristics of the SIB-based architectures are listed in the last three columns of Table I.

TABLE I: CHARACTERISTICS OF THE BENCHMARK SCAN NETWORKS

	MUX-based Architecture			SIB-based Architecture		
Design	Num. MUX	Total #scan regs.	Total #scan bits	Num. SIB	Total #scan regs.	Total #scan bits
f2126	45	81	15 834	41	77	15 830
q12710	30	51	26 188	25	47	26 183
p22810	311	565	30 139	283	537	30 111
p34392	142	245	23 261	123	226	23 242
p93791	653	1241	98 637	621	1209	98 605
t512505	191	319	77 037	160	288	77 006
a586710	47	79	41 682	40	72	41 675

B. Experimental Setup

Optimal restricted accesses are generated with the approach presented in [15] extended with the required constraints, as explained in Section III-A. The resulting sequence filters are synthesized for the Nangate 45nm open cell library [19] with area optimization goal. We calculate the area overhead w.r.t. the area of the scan network, which includes no system logic. The actual area overhead w.r.t. the full chip is expected to be much lower. The resulting operating frequency of all evaluated filters is above 300 MHz, which is significantly more than the usual JTAG clock speed (10 to 100 MHz).

We construct sequence filters for random samples of protected and target scan registers. All results presented in the following sections, including the area overhead and the number of FSM's states, represent average values acquired from the evaluation of 10 filters built for different random samples of target registers. The standard deviation of area overhead is below 1% in the experiments.

C. Filters for Individual Accesses

For each RSN, we construct sequence filters for 10, 20, and 100 restricted accesses to random scan registers. We consider all remaining scan registers which do not control the state of the RSN as protected. The accesses are individual, i.e., each access to a single register is realized by a separate sequence. This is relevant for low-latency access to individual registers.

Table II presents the properties of the sequence filters for individual accesses. The benchmark name, type of scan architecture (MUX/SIB), and benchmark area are given in the first three columns. Columns 4-7 gives the properties of sequence filters that allow 10 restricted accesses: the cumulative length of the sequences, the number of states in the FSM's state diagram, and the filter's area overhead w.r.t. the RSN area (col. 3). The properties of filters allowing 20 and 100 accesses are given in columns 8-11 and 12-15, respectively.

The filter size depends on the number of allowed accesses: For 10 individual accesses, the area overhead ranges from 0.2 to 0.6% (col. 7). For 20 accesses, the area is 0.3 to 1.2% (col. 11), and for 100 accesses it rises up to 4.3% (col. 15). In most cases, the increase in area overhead is less than the increase in the number of allowed accesses. Note that three of our benchmark RSNs (f2126, q12710, and a586710) include less than 100 scan registers (cf. Table I). Even if we allow access to a high fraction or all of their scan registers, the area overhead is still below 2% w.r.t. the RSN (col. 15 in Table II).

The size of a sequence filter is proportional to the number of states in filter's state diagram, which, in turn, is proportional to the length of allowed sequences. Thanks to sequence merging (Section III-C), the number of states in the state diagram is significantly less than the cumulative sequence length: Merging reduces the number of states by 2.4x for 10 accesses to p93791-SIB, and by up to 298x for 100 accesses to q12710-SIB.

D. Filters for Concurrent Accesses

In the second series of experiments we construct sequence filters for the concurrent access to 100 random scan registers, realized by 1, 5, 10 and 20 sequences. The concurrent access is efficient if the registers are usually accessed together.

Table III shows the properties of sequence filters for the concurrent access. In columns 3-5, we describe the filters for a *single* concurrent access to 100 registers, including the sequence length, the number of states in the filter's state diagram, and the area overhead. The properties of filters for 5 accesses à 20 registers, 10 accesses à 10 registers, and 20 accesses à 5 registers are given in the consecutive columns.

For 20 accesses à 5 registers, the filter area overhead (col. 14) is close to the area for individual accesses to these registers (cf. col. 15 in Table II). However, if the access to all 100 registers is combined into a *single* access, the cost (col. 5) is from 3.4x to 12x less than the cost of individual accesses. This is explained by the lower length of the concurrent sequences compared with the cumulative length of individual sequences. Thus, if the registers are often accessed together, the concurrent access is faster, and the filter cost is lower.

V. CONCLUSION

Controllability and observability of scan infrastructure, as required for test and debug, contradict with the security requirements for in-field operation. We propose a scalable approach for restricting the accessibility of reconfigurable scan networks by extending the test access port (TAP) with a sequence filter. The filter permits reconfiguration of the network only with a set of allowed scan-in data sequences. It can be employed for multi-level access management at external and internal interfaces of a scan network. Our approach does not affect the access time, does not require any modification of the network, and is well-suited for core-based designs.

TABLE II: HARDWARE OVERHEAD OF THE SEQUENCE FILTERS FOR INDIVIDUAL ACCESSES

Benchmark			10 Restricted Accesses				20 Restricted Accesses				100 Restricted Accesses			
		Area	Length	States	Area		Length	States	Area		Length	States	Area	
Name *(1)*	Type *(2)*	[μm^2] *(3)*	[cycles] *(4)*	[#] *(5)*	[μm^2] *(6)*	[+%] *(7)*	[cycles] *(8)*	[#] *(9)*	[μm^2] *(10)*	[+%] *(11)*	[cycles] *(12)*	[#] *(13)*	[μm^2] *(14)*	[+%] *(15)*
f2126	SIB	239 902	5 521	345	1 085	+0.45%	10 060	574	1 931	+0.81%	49 192	1 100	3 781	+1.58%
f2126	MUX	240 021	5 009	388	1 247	+0.52%	9 899	663	2 169	+0.90%	51 279	1 317	3 773	+1.57%
q12710	SIB	397 483	12 068	180	751	+0.19%	26 087	274	1 079	+0.27%	125 892	422	1 560	+0.39%
q12710	MUX	397 592	13 242	242	1 003	+0.25%	27 175	368	1 351	+0.34%	131 592	560	2 183	+0.55%
p22810	SIB	453 537	2 436	1 017	2 722	+0.60%	4 993	1 795	4 946	+1.09%	24 096	6 536	19 201	+4.23%
p22810	MUX	454 107	2 258	850	2 435	+0.54%	4 488	1 560	5 283	+1.16%	21 730	5 787	19 297	+4.25%
p34392	SIB	352 290	3 699	678	1 983	+0.56%	7 069	1 263	3 701	+1.05%	32 747	4 247	11 515	+3.27%
p34392	MUX	352 699	3 075	710	2 167	+0.61%	6 216	1 260	3 927	+1.11%	33 475	4 176	12 644	+3.59%
p93791	SIB	1 486 289	3 104	1 287	3 004	+0.20%	6 109	2 432	6 236	+0.42%	31 582	9 892	22 837	+1.54%
p93791	MUX	1 486 772	3 075	1 260	3 316	+0.22%	6 060	2 321	6 318	+0.42%	31 677	9 658	22 360	+1.50%
t512505	SIB	1 167 569	8 063	965	2 294	+0.20%	16 295	1 735	4 537	+0.39%	71 239	5 512	14 998	+1.28%
t512505	MUX	1 168 310	7 161	803	2 194	+0.19%	14 138	1 454	4 361	+0.37%	68 719	4 876	14 323	+1.23%
a586710	SIB	634 087	14 723	318	1 147	+0.18%	28 189	524	1 928	+0.30%	132 255	985	3 701	+0.58%
a586710	MUX	634 258	13 027	377	1 420	+0.22%	27 447	601	2 489	+0.39%	135 388	1 128	3 983	+0.63%

TABLE III: HARDWARE OVERHEAD OF THE SEQUENCE FILTERS FOR CONCURRENT ACCESSES

Benchmark		1 Access to 100 Registers			5 Accesses à 20 Registers			10 Accesses à 10 Registers			20 Accesses à 5 Registers		
		Length	States	Area	Length	States	Area	Length	States	Area	Length	States	Area
Name *(1)*	Type *(2)*	[cycles] *(3)*	[#] *(4)*	[+%] *(5)*	[cycles] *(6)*	[#] *(7)*	[+%] *(8)*	[cycles] *(9)*	[#] *(10)*	[+%] *(11)*	[cycles] *(12)*	[#] *(13)*	[+%] *(14)*
f2126	SIB	15 883	129	+0.21%	44 454	512	+0.84%	43 597	874	+1.29%	47 264	1 491	+2.16%
f2126	MUX	15 907	122	+0.24%	45 334	533	+0.86%	44 997	961	+1.36%	46 573	1 655	+2.47%
q12710	SIB	26 220	82	+0.11%	124 930	292	+0.30%	122 825	604	+0.66%	124 576	959	+0.94%
q12710	MUX	26 237	82	+0.11%	125 138	299	+0.30%	122 744	682	+0.67%	126 538	1 173	+1.08%
p22810	SIB	12 956	752	+0.49%	14 526	2 395	+1.61%	15 967	3 436	+2.31%	16 235	4 664	+2.97%
p22810	MUX	15 721	688	+0.42%	15 765	2 291	+1.55%	14 500	3 216	+2.12%	16 311	4 368	+2.67%
p34392	SIB	22 667	448	+0.29%	24 524	1 454	+1.31%	25 337	2 428	+2.07%	27 549	3 798	+3.34%
p34392	MUX	24 240	449	+0.30%	22 491	1 613	+1.53%	25 685	2 699	+2.35%	26 145	4 100	+3.78%
p93791	SIB	18 518	1 518	+0.27%	21 580	5 264	+0.77%	24 026	7 423	+1.17%	25 908	9 303	+1.34%
p93791	MUX	31 162	1 227	+0.19%	25 471	4 792	+0.72%	24 017	6 987	+1.11%	25 344	8 643	+1.50%
t512505	SIB	58 833	455	+0.11%	57 912	1 341	+0.38%	61 220	2 140	+0.60%	62 592	3 368	+0.91%
t512505	MUX	59 635	482	+0.10%	62 407	1 415	+0.36%	62 371	2 100	+0.57%	63 722	3 194	+0.81%
a586710	SIB	41 748	144	+0.10%	130 674	529	+0.34%	127 817	887	+0.51%	126 344	1 416	+0.84%
a586710	MUX	74 397	148	+0.10%	130 097	707	+0.45%	129 003	1 197	+0.76%	133 670	1 846	+1.03%

Experimental results show that in average, to assure security and retain the accessibility of 100 scan registers, the proposed approach increases the area of scan infrastructure by less than 5%, which is marginal with respect to the chip area.

ACKNOWLEDGEMENTS

This work was supported by the German Research Foundation (DFG) under grants WU 245/13-1 (RM-BIST) and WU 245/11-1 (OASIS).

REFERENCES

[1] N. Stollon, *On-Chip Instrumentation: Design and Debug for Systems on Chip*. Springer US, 2011.

[2] E. Larsson and K. Sibin, "Fault management in an IEEE P1687 (IJTAG) environment," in *IEEE Int'l Symposium on Design and Diagnostics of Electronic Circuits Systems (DDECS)*, 2012, p. 7.

[3] J. Rearick and A. Volz, "A Case Study of Using IEEE P1687 (IJTAG) for High-Speed Serial I/O Characterization and Testing," in *Proc. IEEE International Test Conference (ITC)*, 2006, paper 10.2.

[4] J. Da Rolt, G. Di Natale, M.-L. Flottes, and B. Rouzeyre, "Are advanced DfT structures sufficient for preventing scan-attacks?" in *Proc. IEEE VLSI Test Symposium (VTS)*, 2012, pp. 246–251.

[5] M. Tehranipoor and C. Wang, *Introduction to Hardware Security and Trust*. Springer, 2011.

[6] B. Yang, K. Wu, and R. Karri, "Scan Based Side Channel Attack on Dedicated Hardware Implementations of Data Encryption Standard," in *Proc. Int'l Test Conf. (ITC)*, 2004, pp. 339–344.

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

[8] K. Park, S. Yoo, T. Kim, and J. Kim, "JTAG Security System Based on Credentials," *Journal of Electronic Testing*, vol. 26, pp. 549–557, 2010.

[9] K. Rosenfeld and R. Karri, "Attacks and Defenses for JTAG," *IEEE Design & Test of Computers*, vol. 27, no. 1, pp. 36–47, Jan.-Feb. 2010.

[10] Y. Shi, N. Togawa, M. Yanagisawa, and T. Ohtsuki, "Robust Secure Scan Design Against Scan-Based Differential Cryptanalysis," *IEEE Trans. VLSI Syst.*, vol. 20, no. 1, pp. 176–181, Jan. 2012.

[11] K. Rosenfeld and R. Karri, "Security-Aware SoC Test Access Mechanisms," in *Proc. IEEE VLSI Test Symp. (VTS)*, 2011, pp. 100–104.

[12] E. Ebrard, B. Allard, P. Candelier, and P. Waltz, "Review of fuse and antifuse solutions for advanced standard CMOS technologies," *Microelectronics Journal*, vol. 40, no. 12, pp. 1755 – 1765, 2009.

[13] L. Sourgen, "Security Locks for Integrated Circuit," May 1992, US Patent App. 5101121 A.

[14] R. Baranowski, M. A. Kochte, and H.-J. Wunderlich, "Modeling, Verification and Pattern Generation for Reconfigurable Scan Networks," in *Proc. IEEE Int'l Test Conf. (ITC)*, 2012, paper 8.2.

[15] R. Baranowski, M. A. Kochte, and H.-J. Wunderlich, "Scan Pattern Retargeting and Merging with Reduced Access Time," in *Proc. IEEE European Test Symposium (ETS)*, 2013.

[16] M. Nicolaidis, S. Noraz, and B. Courtois, "A Generalized Theory of Fail-Safe Systems," in *Int'l Symp. on Fault-Tolerant Computing (FTCS), Digest of Papers*, 1989, pp. 398–406.

[17] E. Marinissen, V. Iyengar, and K. Chakrabarty, "A Set of Benchmarks for Modular Testing of SOCs," in *Proc. Int'l Test Conf. (ITC)*, 2002, pp. 519–528.

[18] F. Zadegan, U. Ingelsson, G. Carlsson, and E. Larsson, "Design Automation for IEEE P1687," in *Proc. Design, Automation Test in Europe Conf. (DATE)*, 2011, pp. 1412–1417.

[19] Nangate 45nm Open Cell Library v1.3, http://www.nangate.com.