Access Port Protection for Reconfigurable Scan Networks

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

Journal of Electronic Testing: Theory and Applications (JETTA) Vol. 30(6) December 2014

url: http://link.springer.com/article/10.1007/s10836-014-5484-2

doi: https://doi.org/10.1007/s10836-014-5484-2

Abstract: Scan infrastructures based on IEEE Std. 1149.1 (JTAG), 1500 (SECT), and P1687 (IJTAG) provide a cost-effective access mechanism for test, reconfiguration, and debugging purposes. The improved accessibility of on-chip instruments, however, poses a serious threat to system safety and security. While state-of-theart protection methods for scan architectures compliant with JTAG and SECT are very effective, most of these techniques face scalability issues in reconfigurable scan networks allowed by the upcoming IJTAG standard. This paper describes a scalable solution for multilevel access management in reconfigurable scan networks. The access to protected instruments is restricted locally at the interface to the network. The access restriction is realized by a sequence filter that allows only a precomputed set of scan-in access sequences. This approach does not require any modification of the scan architecture and causes no access time penalty. Therefore, it is well suited for core-based designs with hard macros and 3D integrated circuits. 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 registers in the network.

Journal of Electronic Testing: Theory and Applications (JETTA) manuscript No.
(will be inserted by the editor)

Access Port Protection for Reconfigurable Scan Networks

Rafal Baranowski · Michael A. Kochte · Hans-Joachim Wunderlich

Received: date / Accepted: date

Abstract Scan infrastructures based on IEEE Std. 1149.1 (JTAG), 1500 (SECT), and P1687 (IJTAG) provide a cost-effective access mechanism for test, reconfiguration, and debugging purposes. The improved accessibility of on-chip instruments, however, poses a serious threat to system safety and security. While state-of-the-art protection methods for scan architectures compliant with JTAG and SECT are very effective, most of these techniques face scalability issues in reconfigurable scan networks allowed by the upcoming IJTAG standard.

This paper describes a scalable solution for multi-level access management in reconfigurable scan networks. The access to protected instruments is restricted locally at the interface to the network. The access restriction is realized by a *sequence filter* that allows only a precomputed set of scan-in access sequences. This approach does not require any modification of the scan architecture and causes no access time penalty. Therefore, it is well suited for core-based designs with hard macros and 3D integrated circuits. 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 registers in the network.

Keywords Debug and diagnosis · reconfigurable scan network · IJTAG · IEEE P1687 · secure DFT · hardware security

This manuscript is an extended version of the paper [6] presented at the 22nd IEEE Asian Test Symposium, Nov. 2013.

This work was supported by the German Research Foundation (DFG) under grants WU 245/13-1 (RM-BIST) and WU 245/17-1 (ACCESS).

R. Baranowski · M.A. Kochte · H.J. Wunderlich
Institut für Technische Informatik, Universität Stuttgart
Pfaffenwaldring 47, D-70569 Stuttgart, Germany
Tel.: +49 711 685 88 362, Fax: +49 711 685 88 288
E-mail: baranowski@iti.uni-stuttgart.de

M.A. Kochte
E-mail: kochte@iti.uni-stuttgart.de

H.J. Wunderlich
E-mail: wu@iti.uni-stuttgart.de

1 Introduction

To facilitate cost-effective VLSI development and improve product dependability, VLSI designs incorporate on-chip instrumentation that makes the process of production ramp-up more tractable and facilitates in-field system maintenance. This embedded instrumentation includes, for instance, debug structures for post-silicon validation, scan chains for test and diagnosis, as well as components that enable in-field system monitoring, reconfiguration, diagnosis, and repair [28, 1, 33, 3].

Embedded instruments are usually integrated into the system-wide scan infrastructure and accessed via the four-wire Test Access Port (TAP) defined by IEEE Std. 1149.1 (Joint Test Action Group, JTAG [17]). Over the years, the TAP interface has become the *de facto* standard for efficient, low-cost access to on-chip instrumentation [22, 33].

Scan architectures based exclusively on JTAG do not scale well with the number of instruments and hence are insufficient for efficient access to instrumentation embedded in complex System-on-a-Chip (SoC) designs [19, 15]. To improve access flexibility and reduce access time, various *Reconfigurable Scan Network* (RSN) architectures have been proposed. In such architectures, scan cells or instruments that need not be accessed can be temporarily excluded from the scan chain [19]. The path through which data are shifted

in an RSN is configured by the state of *configuration registers* embedded in the RSN itself. Such architectures emerge as a scalable option for the access to on-chip instrumentation, offering a flexible, low-latency, and low-cost access [28, 33, 19, 3]. The ongoing effort IEEE Std. P1687 (or IJTAG for *Internal* JTAG) aims to standardize the design and access to this type of scan networks [29, 14, 33].

System-wide scan infrastructure often provides the access to sensitive instrumentation and is therefore prone to abuse, sabotage, unlicensed usage, or intellectual property (IP) theft [34, 10]. An attacker may exploit the scan infrastructure to gain access to protected data (secret key or IP), alter the system state by fault injection, or perform illegal operations. Successful attacks on the JTAG interface are reported for pirating satellite TV services, circumventing mechanisms for Data Rights Management (DRM) [34], or retrieval of secret keys from cryptographic cores [35].

Different levels of infrastructure accessibility are required during product development, volume production, and in-field operation. For instance, in production ramp-up, volume test and diagnosis, high observability and controllability is key to low time to market and high product quality. However, during in-field operation and maintenance, the accessibility of chip internals must be restricted due to security and safety reasons, e.g. to prevent IP theft or tampering. In automotive applications, for instance, full access is mandatory during manufacturing and assembly test, while only limited access is allowed during operation and maintenance in a workshop to prevent unauthorized chip tuning.

The IEEE 1149.1 Test Access Port (TAP) can be protected against unauthorized access using authentication mechanisms, scan data encryption, and access restriction methods [30, 10]. Such techniques can be directly applied to protect RSNs which are integrated as JTAG data registers. With this approach, however, an RSN is protected as a whole, and fine-grained security control over individual instruments is impossible.

This paper presents a novel protection method that offers scalable, multi-level access management for reconfigurable scan networks. This method is based on *sequence filters* that are placed locally at the JTAG TAP, as shown in Figure 1). A sequence filter prevents the access to a specified set of *protected instruments* and allows *restricted access* to remaining (unprotected) instruments. The latter instruments remain accessible and hence are not protected by the filter. In contrast to the majority of state-of-the-art protection techniques, sequence filters can be applied to arbitrary RSNs, require no modification of the RSN architecture, and need no additional global wiring for security control. For this reason, this protection method is well suited for core-based designs with hard macros and 3D integrated circuits. It is also directly applicable to complex RSNs compliant with IEEE P1687 and requires just a high-level network description in Instrument Connectivity Language (ICL [33]), or a precomputed set of scan-in data for allowed accesses.

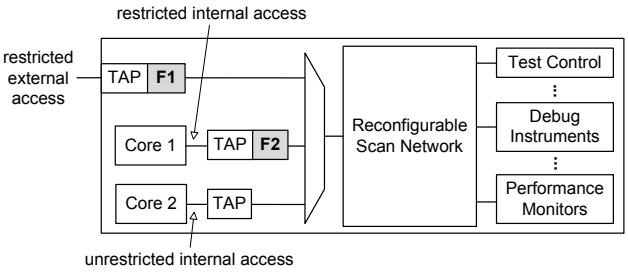


Fig. 1: Example of a multi-level access port protection based on *sequence filters* (F1, F2)

A sequence filter is activated either statically with an on-chip fuse, e.g. after manufacturing test, or deactivated dynamically for authorized users. An example of a *multi-level* filter-based protection for a System-on-a-Chip (SoC) is given in Figure 1. Filter F1 restricts the external accessibility of debug instrumentation, e.g. for IP protection. F2 blocks the internal accessibility of embedded test instruments at the TAP of "Core 1", e.g. due to safety requirements for in-field operation. Still, full internal accessibility is preserved for debugging purposes via the internal TAP of "Core 2". Apart from blocking the access to protected instruments, sequence filters can also be used to allow sequential access to a set of instruments, and still block simultaneous (concurrent) access to them, e.g. to prevent that sensitive data are shifted through exposed or untrusted instruments.

In the next section, we give a short introduction to reconfigurable scan networks. Section 3 discusses state-of-the-art protection techniques for scan infrastructure and compares them with the proposed approach. An overview of the protection method is given in Section 4, followed by the detailed method for the generation of restricted accesses (Section 5) and for the synthesis of sequence filters (Section 6). The hardware overhead of sequence filters is evaluated in Section 7.

2 Reconfigurable Scan Networks

This work deals with the protection of reconfigurable scan networks (RSNs) as defined in the upcoming IEEE Std. P1687 and the novel IEEE Std. 1149.1-2013. A

simplified description of the structure and functionality of such networks is presented below. For a more detailed introduction please refer to [3].

RSNs are usually accessed through a JTAG-compliant Test Access Port (TAP) [17] and can be viewed as a scan register with *variable length*. The logic state of the RSN determines which registers (instruments) in the network are currently accessible. The RSN state may be changed by rewriting the content of accessible registers.

RSNs can be decomposed into basic components, such as scan registers, multiplexers, or combinational logic blocks. The basic building block of an RSN is a *scan segment*, as shown in Figure 2. In the simplest case, a scan segment is a shift register composed of one or more *scan flip flops* sharing a set of control signals. A scan segment may possess a *shadow* register, e.g. for bidirectional communication with an on-chip instrument. A scan segment supports up to three operations: During a *capture* operation, the shift register may be loaded with data from an attached instrument. During a *shift* operation, data are shifted from the segment's scan-input, through its register bits, down to the scan-output. During an *update* operation, the optional *shadow* register is loaded with data from the shift register. The shadow register is stable during the shift operation (as in JTAG test data registers). Optionally, a scan segment may also possess a *select* control port, which specifies if the segment is enabled for the capture, shift, and update operation. The optional elements are dashed in Figure 2.

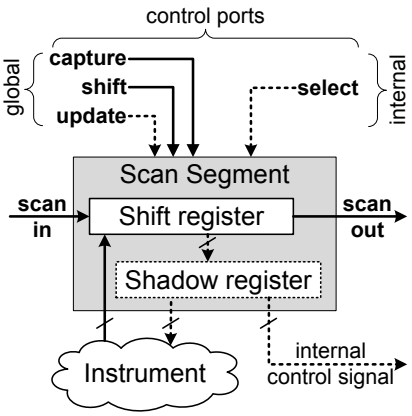


Fig. 2: Scan segment

An RSN may include *scan multiplexers*, i.e. multiplexers which control the path through which scan data are shifted in the network. The control port of a scan multiplexer is called *address* and specifies the selected scan input.

The state of the internal control ports, such as *select* or *address*, depends on the logic state of the RSN itself: These ports may be driven by arbitrary combinational logic blocks that take their input from shadow registers of scan segments distributed in the RSN.

An RSN has a *primary scan-input* and a *primary scan-output*, a *clock* input, as well as three *global control inputs* that activate the scan operations: *capture*, *shift*, and *update*. The global control inputs are distributed to all scan segments. If the RSN is accessed through a JTAG TAP, these signals are driven by the TAP controller.

A *scan path* is a non-circular sequence of daisy-chained scan segments starting at the primary scan-in port and ending at the primary scan-out port. A scan path is *active* if and only if the *select* signals for all on-path scan segments are asserted and all on-path multiplexers select the inputs that belong to the active scan path. The state of all scan segments in the RSN and the resulting configuration of the active scan path is referred to as *scan configuration*.

The basic access to a scan network consists of three phases, as defined by IEEE Std. 1149.1 [17]: *capture, shift, and update* (CSU, cf. Figure 3). During capture, the shift registers on the active scan path may latch new data. These data are shifted out during the shift phase, while new scan data are shifted in. Finally, during the update phase, the shifted-in data are latched in the (optional) shadow registers on the active scan path. A read or write access to a scan segment in the network requires that the accessed segment is part of the active scan path. A *scan access* is a sequence of CSU operations required to reconfigure the scan network and access the target registers. An *access pattern* is the scan-in data (sequence of bits) that are applied to the network during a scan access.

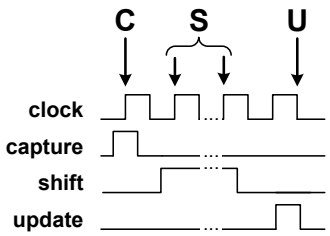


Fig. 3: Capture, Shift, Update (CSU) operation

An example of a simple RSN is given in Fig. 4. The one-bit scan segments `S1` and `S3` contain shadow reg-

isters that drive internal control signals, i.e. segment *select* and multiplexer *address* signals. The state of the shadow registers in S1 and S3 determines the accessibility of two multi-bit scan segments S2 and S4, respectively. In the initial scan configuration, it is assumed that S1 = S3 = 0, hence both S2 and S4 are initially bypassed. The scan-in data is shifted through segments S2 and S4 only if the previous access assured that S1 = S3 = 1. In later examples, we assume that the access to scan segment S2 is allowed, while S4 is *protected*, i.e., S4 must never be put on the active scan path.

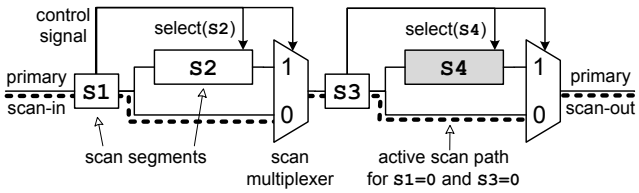


Fig. 4: Example of a reconfigurable scan network with a *protected* scan segment S4

3 Related Work

In the following, state-of-the-art techniques for the protection of scan infrastructure are briefly reviewed and their applicability to reconfigurable scan networks is discussed. For a more detailed introduction to scan protection techniques please refer to the recent survey in [10].

To guarantee inaccessibility of protected instruments, the physical interface to the scan network or parts of the scan infrastructure can be made permanently unusable once the instrumentation is not needed anymore. In the simplest scenario, the entire JTAG TAP is removed after manufacturing test with a wafer saw [18]. This radical approach results in high security but makes the scan infrastructure completely unusable. This is not acceptable in modern SoC designs where at least limited access to instrumentation must be provided throughout the lifetime of a chip.

Alternatively, elements of a scan infrastructure can be deactivated using One Time Programmable (OTP) memory cells called on-chip fuses [13]. By blowing a fuse, some instructions of the JTAG TAP controller or chosen scan chains can be permanently disabled [32]. Most often, the fuses are blown after manufacturing test to prevent that scan chains are used for side-channel attacks on cryptographic cores or theft of intellectual property [34]. Such fuse-based protection is widely adopted in microprocessors, e.g. in i.MX31 (Freescale) [34] or MPS430 (Texas Instruments) [9].

To prevent that sensitive data are revealed by scan infrastructure, the scan data can be protected by encryption. On-chip stream ciphers are used to decrypt the scan-in bit sequence and encrypt the scan-out bit sequence at the JTAG TAP interface [30]. This encryption scheme effectively prevents sniffing and spoofing of secret data at the TAP level. To prevent that scan data are shifted in plaintext through untrusted on-chip instruments or cores, the encryption circuitry can be distributed over the chip to locally decrypt scan-in data and encrypt scan-out data of individual network components [31]. However, if many components require protection, this scheme becomes unwieldy and incurs high hardware overhead.

To account for distinct access rights of different authorized entities, authentication mechanisms are required: The user (e.g. a tester or a service person) gains permission to access the scan network only after proving its identity, e.g. by providing a password or *key*. The simplest authentication schemes assume a static key that is only known to entitled users. To gain access, the key must be either applied to dedicated primary inputs [16], embedded at constant [20, 2] or variable [12] positions in scan data (scan-in bit sequence), or written to a dedicated data register in a JTAG circuitry [21] or an IEEE 1500 wrapper [8]. Since the secrets are distributed to all authorized entities and transported to the chip in plaintext, the probability that such protection schemes are eventually compromised by secret leakage is usually too large for systems with basic security requirements.

Stronger authorization schemes are based on challenge-response protocols [7, 9, 30, 26, 11, 27]: The chip generates a non-repeating *challenge* value and expects the user to provide the expected *response* value based on a *shared secret*. This shared secret is never transferred in plaintext (unencrypted) during the authorization process. The *response* is calculated from the *challenge* using various cryptographic algorithms, e.g. elliptic curve arithmetic [7, 11] or hash functions [9, 30]. More advanced schemes require mutual authentication based on three-entity protocols that require certification authorities or authentication servers [25, 26, 11].

The above-mentioned access restriction techniques and authorization mechanisms can be extended in a straightforward way to protect chosen RSN components: For instance, an authentication controller or an on-chip fuse can be used to force the *select* signal of protected instruments to 0 for unauthorized users. This simple solution, however, does not scale well: Each instrument either needs a dedicated fuse or a local au-

thentication controller, which entails high hardware cost, or must be connected to a central authorization controller or OTP memory, which may result in high routing overhead and routing congestion. In both cases, the original scan network design requires modification, and the protection needs to be considered at early design stages.

To our best knowledge, the only existing protection technique suitable for large RSNs has been recently proposed in [12]. In this technique, protected instruments are accessible via programmable gateways called *Locking Segment Insertion Bits* (LSIB). An LSIB is open only after a predefined multi-bit *key* is loaded into shift registers that may be distributed over the entire RSN. This way, each instrument can be protected individually. This technique requires that the original design of the RSN and possibly the access mechanism be modified. For each protected instrument, this approach entails either the area overhead for the sequential elements that store the multi-bit key, or—if these elements are shared with system logic—the routing overhead. Both the hardware overhead and the access time overhead are proportional to the number of protected instruments, or the number of distinct keys. Since the key or secret is exposed while communicating with the chip and must be known to all authorized entities, this approach is most effective if the requirements on system security are relatively low.

Compared with the existing protection methods, our filter-based technique has unique properties: It does not change the access mechanism and requires only a local modification to the JTAG Test Access Port (TAP). Therefore, it is well suited for core-based designs with hard macros. Since no additional global wires are required, our approach does not cause any routing issues and requires no additional Through Silicon Vias (TSV) in 3D integrated circuits. Finally, as the area overhead is proportional to the number of *unprotected* instruments, the filter-based technique is favorable when the majority of instruments in the system need protection. Our approach is therefore complementary to techniques with hardware overhead proportional to the number of *protected* instruments, such as the LSIB-based approach [12].

4 Protection Overview

The aim of the proposed protection method is to prevent access to protected scan segments by allowing *restricted access patterns* only. An access to the RSN is restricted if it does not put any protected scan segment on the active scan path. For instance, assuming that scan segment `S4` in Figure 4 is protected, restricted access to the target scan segment `S2` must ensure that `S4` is bypassed at all times. To this end, `S3` must always be loaded with 0.

Restricted accesses are enforced with a *sequence filter* that is placed between the TAP and the scan network, as shown in Figure 5. The filter observes the sequence of scan operations (*capture*, *shift*, *update*) and the scan data at the TDI port to decide whether the access pattern is allowed or forbidden. If the scan operations do not expose any protected scan segment, the filter does not interfere with the access (*allow* signal is set to 1). Otherwise, the filter inhibits the *update* operation (*allow* set to 0) and so prevents all RSN registers from latching any data that could expose or give access to any protected scan segment.

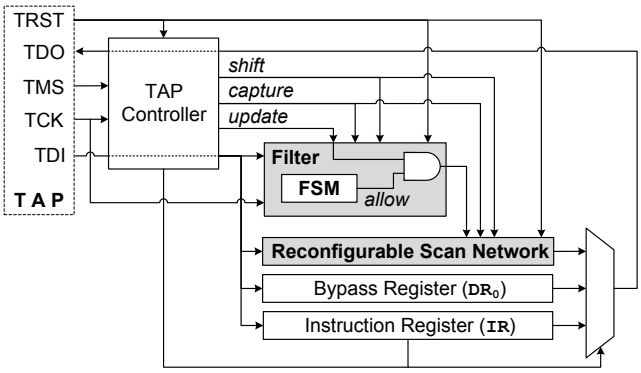


Fig. 5: Test Access Port (TAP) protection using a *sequence filter*

A single sequence filter can be used to allow any number of restricted access patterns, whereby each restricted access can be enabled or disabled separately to allow multi-level access management. A filter can be activated using a fuse, e.g. after manufacturing test or before the complete system is delivered to the customer. Alternatively, using an authentication mechanism, restricted accesses supported by the filter can be dynamically enabled for authorized users. Preferably, challenge-response protocols should be used to prevent that secret data are leaked during the authentication process. The challenge-response authentication can be performed over the JTAG TAP interface, as described for instance in [7, 9, 26, 11].

An overview of the proposed method is presented in Figure 6. The restricted access patterns are generated in an automated way for a given set of target and protected scan segments, as discussed in Section 5. The restricted patterns are fed to the filter synthesis algorithm presented in Section 6.

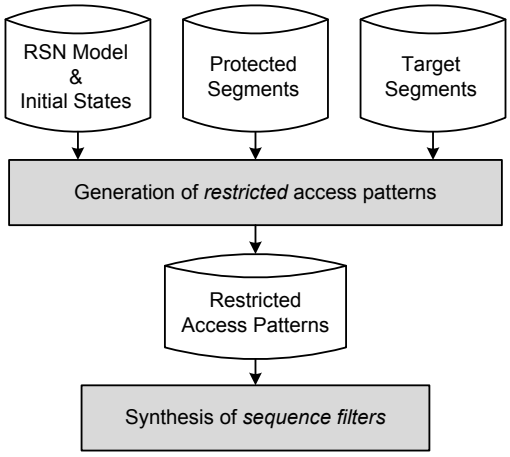


Fig. 6: Overview of the proposed protection method

4.1 Security Analysis

Sequence filters protect the scan infrastructure against non-invasive attacks that involve the observation and controlling of chip-internal and -external interfaces. If a sequence filter is enabled statically by an on-chip fuse, the security of protected scan segments is comparable to the security achieved by interface (TAP) removal or direct instrument deactivation using locally placed fuses. If a sequence filter is deactivated dynamically for authorized users, the security level depends on the implemented authentication protocol and the security of cryptographic hardware primitives.

To provide protection against fault-injection attacks, the design must be additionally equipped with sensors that can detect under/over voltage, extreme temperatures, as well as clock and reset instability or glitches [34]. If any abnormality is detected, the *allow* signal must be set to 0. Preventive actions against invasive attacks that involve chip dismantling, reverse-engineering, and microprobing are reviewed in [34].

To guarantee security in presence of soft errors and hardware defects, sequence filters can be designed fail-safe [24]: In presence of faults, the output *allow* must be either correct or inactive (0). For instance, to protect against single faults, the filter is duplicated and the *allow* outputs are used as a dual-rail encoded signal, or conjoined with an AND gate.

4.2 Testability Analysis

Since a sequence filter is in principle a finite state machine, standard design-for-test techniques can be used to improve its testability: The sequential elements of the filter can be made scannable, and an additional scan cell can be used to observe the *allow* output. The scan chain composed of the filter's scan cells can be interfaced as a separate data register with the JTAG TAP. Care must be taken to assure that the protected RSN becomes inaccessible as soon as the filter enters the test mode. To this end, the *allow* signal must be forced to 0 when the filter's scan chain is accessed. This signal can only be released after both the filter and the protected RSN undergo a reset.

5 Restricted Access Generation

In the following, restricted access patterns are defined formally:

Definition 1 (Restricted Access Pattern) For a given RSN design, let S be the set of scan segments, C set of scan configurations, and $c_0 \in C$ the initial (reset) scan configuration. Given a set of *protected* segments $S_P \subset S$ and a set of initial scan configurations $I \subseteq C$ such that $c_0 \in I$, an access pattern for *target* scan segments in $S \setminus S_P$ is *restricted* if it fulfills all of the following conditions:

- The target segments are properly accessed for all initial scan configurations in I.
- During the access, no protected scan segment from S_P belongs to the active scan path (the scan data do not pass through any protected scan segment).
- For all initial scan configurations in I, the scan configuration after the access belongs to set I.

Since the final scan configuration after a restricted access belongs to the set of initial scan configurations I, it follows from Definition 1 that any concatenation of restricted accesses is also a restricted access.

Unrestricted access patterns with *minimal access time* are generated in an automated way using the RSN modeling and pattern generation method presented in [5]. This method maps the pattern generation problem to a Boolean satisfiability problem by constructing a Boolean formula (*SAT instance*) that is satisfiable if and only if an access pattern exists that reads or writes the set of target scan segments within a given number of CSU operation. The satisfying assignment to this formula provides the access pattern (scan data).

To generate *restricted* access patterns, the method from [5] is extended in the following way: The SAT instance representing a restricted access with n CSU

operations is constructed as follows:

$$\texttt{Access}(n) := \Omega_I(V_0) \wedge \left[\bigwedge_{i=1\ldots n} \Omega_T(V_{i-1}, V_i)\right] \wedge \Omega_R(V_0, V_1, \ldots, V_n) \wedge \underbrace{\Omega_I(V_n) \wedge \left[\bigwedge_{i=0\ldots n} \bigwedge_{s\in S_P} \neg\texttt{Active}(V_i, s)\right]}_{\text{constraints for restricted access}},$$

where for $0 \leq i \leq n$, V_i denotes the set of state variables for the i-th scan configuration (after applying the i-th CSU operation), Ω_R represents access constraints for the target scan segments in the final and/or intermediate scan configurations, Ω_I and Ω_T are the characteristic functions of the set of initial scan configurations I and of the transition relation of the RSN model, respectively, and $\texttt{Active}(V_i, s)$ is a predicate that holds if and only if the scan segment s belongs to the active scan path in the i-th scan configuration.

This instance is satisfiable if and only if there exists a restricted access with n CSU operations such that target scan segments are properly accessed (Ω_R is satisfied), protected scan segments in S_P never belong to the active scan path (their content is never altered nor exposed), and the initial scan configuration I is restored. For the details on RSN modeling and access pattern generation please refer to [5, 3].

The proposed method poses two reasonable requirements on the RSN: (1) In the initial (reset) scan configuration, no protected scan segment may belong to the active scan path. (2) There must exist a way to bypass all protected scan segments while accessing target scan segments. If the access to a target segment requires that any protected scan segment be modified or exposed, the protected segment needs to be extended with a configurable bypass that is initially active, e.g. a Segment Insertion Bit (SIB) [33, 15].

5.1 Restricted Access Example

In the RSN from Figure 4, scan segment S4 is protected while the access to segment S2 is allowed. Assume that I is defined as the set of all scan configurations in which $\texttt{S1} = \texttt{S3} = 0$. According to Definition 1, a restricted access to S2 must guarantee that:

- S2 is accessed for all 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 pattern for segment S2 consists of two CSU operations with the following scan data (leftmost bit is shifted first): *01* and *0X0*, where *X* stands for the target value of S2. The first CSU operation puts segment S2 on the active scan path by setting S1 to 1. In the second CSU, S2 is accessed and the initial state of S1 is restored. During the two CSU operations, the protected segment S4 is bypassed. After the access, the final scan configuration satisfies $\texttt{S1} = \texttt{S3} = 0$.

6 Sequence Filter Synthesis

A sequence filter consists of a Finite State Machine (FSM) that receives the scan data input (TDI) of the TAP, as well as the *capture*, *shift*, and *update* control signals driven by the TAP controller (cf. Figure 5). Optionally, the filter may have additional inputs for controlling the *access level*, i.e. inputs enabling a specified subset of restricted access patterns. The state diagram of the filter's FSM is constructed directly from a set of user-defined restricted access patterns, as described below. The FSM tracks scan operations at the TAP and generates a single output *allow* which controls the *update* operation in the RSN: As long as the sequence of scan operations matches any allowed restricted access, the *allow* signal is active and the access is applied to the RSN without any delay. Otherwise, *allow* is deactivated and the FSM enters a *trap* state. In the trap state, no further reconfiguration of the RSN is allowed, and hence no access to protected scan segments is possible.

The state of the filter's FSM must be synchronized with the scan configuration of the protected RSN. This requires that two conditions hold: (1) The reset signal reliably puts both the RSN and the sequence filter to their initial states. (2) While the sequence filter is in operation, the scan segments that control the active scan path (*configuration segments*) are only accessible via the protected TAP. If the RSN is accessed through another TAP (e.g. via an internal interface) or the state of configuration segments is changed internally in the system due to any other reason, the sequence filter must be put into the *trap* state. This assures that no forbidden access can take place when the sequence filter is not synchronized.

6.1 State Diagram Construction

Procedure 1 presents the state diagram construction algorithm for sequence filters. The input to the procedure is a set of sequences (strings) representing restricted accesses patterns that the filter should allow

(`sequenceSet`). The input sequences are composed of five scan operations denoted as follows:

- *0*: shift of bit 0,
- *1*: shift of bit 1,
- *X*: shift of an unconstrained (*don't care*) bit,
- *C*: capture,
- *U*: update.

For instance, a restricted access consisting of two CSU operations with scan data *01* and *0X0* is represented by the following sequence: *C01UC0X0U*. Note that a single sequence represents 2^k restricted access patterns, where k is the number of unconstrained data bits (*X*) in the sequence.

Procedure 1 State diagram construction for sequence filters

```
Input: sequenceSet
Output: state diagram
 1: Create initialState, trapState.
 2: Annotate initialState with all sequences from sequenceSet.
 3: currentStateSet ← {initialState}
 4: n ← 0
 5: while currentStateSet ≠ ∅ do
 6:     nextStateSet ← ∅
 7:     for all state ∈ currentStateSet do
 8:         for all sequence ∈ annotations of state do
 9:             transition ← sequence[n]
10:             if transition = U and length(sequence) = n + 1 then
11:                 Add transition from state to initialState.
12:             else
13:                 Create newState and annotate it with sequence.
14:                 Add transition from state to newState.
15:                 Add newState to nextStateSet.
16:             end if
17:         end for
18:     end for
19:     Replace overlapping transitions from states in
20:         currentStateSet.
21:     Add escape transitions from states in
22:         currentStateSet to trapState.
23:     Merge equivalent states in nextStateSet.
24:     currentStateSet ← nextStateSet
25:     n ← n + 1
26: end while
27: Collapse state sequences with equivalent outbound transitions.
```

The diagram construction algorithm starts with the creation of an "initial" state (`initialState`) that corresponds to the set of initial scan configurations, and a "trap" state (`trapState`) that is reached upon detection of any forbidden scan operation (line 1 in Procedure 1). Each state in the state diagram is annotated with the sequences that put the FSM into this state. State transitions are conditioned either by a single 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 as *C*,*U*). All states are stable as long as no scan operation takes place.

The construction algorithm is a stepwise procedure (lines 5 to 26): In the first step, the first scan operation of each sequence is processed (i.e., the capture operations). In the n-th step, another level of states is added to the state diagram based on the n-th scan operations of the provided sequences (lines 13 to 15). The current scan operation in each sequence is assigned a new successor state (`newState`) with an incoming transition from the respective state in `currentStateSet`. Since any concatenation of restricted accesses is also a restricted access (cf. Section 5), the last *update* operation in each sequence corresponds to a transition to the initial state (line 11). 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 (line 19): 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 Figure 7.

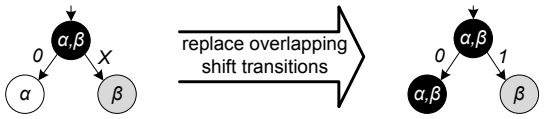


Fig. 7: Example of a state diagram before and after replacement of overlapping shift transitions. Annotations α and β denote two sequences.

After execution of line 19, if all scan operations are allowed in a state from `currentStateSet`, this state has either 3 or 4 outbound transitions, conditioned by either *C*, *U*, and *X* or *C*, *U*, *0*, and *1*. If some operations are *forbidden* (not allowed by any provided sequence), the sequence filter must detect them and prevent any further reconfiguration of the network. To this end, an *escape transition* pointing to the `trapState` is added for the forbidden operations (line 21). Once a forbidden operation is encountered, the filter is stuck in the `trapState`. In this state, the update operation is inhibited until the sequence filter and the scan network are reset.

Optionally, to support multi-level access management, an additional transition to the `trapState` can be added for each restricted access pattern that should be enabled or disabled dynamically. Such transitions must be conditioned by external inputs specifying the current *access level*.

6.2 State Merging and Sequence Collapsing

To reduce the size of the state diagram, redundancies are removed by *merging* equivalent states (line 23 in

Procedure 1) and *collapsing* sequences of states with equivalent outbound transitions (line 27), as described below.

Each pair of successor states in `nextStateSet` is *merged* into a single state if it fulfills one of the following conditions:

- The two states have identical annotations (belong to the same sequences).
- The inbound transitions of the two states have the same condition, and their predecessors have the same annotations.

A state that results from merging of 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) bits (see example in Figure 9a). Typically, long sequences of X operations represent unconstrained data for scan segments that do not control the active scan path. Such sequences are *collapsed* into a single state, and a counter is used to keep track of their length, as shown in Figure 8. During a transition to a collapsed state, the counter is set to the number of states that were removed due to collapsing (via the *value* signal; by asserting the *load* signal). The counter is decremented upon detection of every shift transition (via the *decrement* input) and asserts its *wait* output as long as its value is larger than zero. The FSM leaves the collapsed state as soon as the *wait* signal is deasserted or a forbidden operation (C or U) is detected. Just a single counter is required regardless of how many sequences are collapsed.

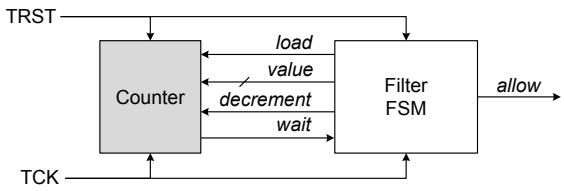


Fig. 8: Sequence filter augmented with a counter for collapsed states

Figure 9 presents an example for collapsing the sequence *XXX1*. The states b, c, d in Figure 9a are collapsed into a single state m in Figure 9b. During the transition to the collapsed state m, the counter is set to 2. The counter is decremented upon every shift operation (X). The final state e is reached as soon as the *wait* signal is deasserted and the final scan operation is correct (*1*). Otherwise, the trap state is reached.

The final state diagram can be further optimized to allow repeated accesses to a set of target scan segments

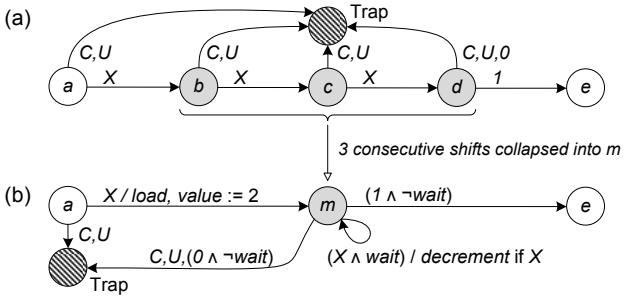


Fig. 9: Example for sequence collapsing with (a) a state diagram and (b) its collapsed equivalent

with little or no hardware overhead. This is crucial to apply many patterns to a set of scan segments with no access time penalty for scan path reconfiguration. To this end, just two repeated accesses must be reflected in the input sequence, such that the first access does not modify the active scan path. The resulting state diagram is then extended with a loop transition for the first access, which enables an unlimited number of repeated accesses. This is explained at an example below.

6.3 Sequence Filter Example

In the following, the sequence filter construction is illustrated at the example of the RSN from Figure 4. The filter is constructed for two restricted accesses characterized by the sequences α and β:

- α: *C01UC0X0U*, which accesses `S2` once (as in Section 5.1),
- β: *C01UC0X1UC0X0U*, which accesses `S2` twice.

Such sequences are found in an automated way using the approach presented in Section 5.

Figure 10 presents the state diagram constructed for the sequences α and β by Procedure 1. The annotations of states are denoted inside the state symbols (α and β). For the sake of clarity, the escape transitions to the trap state are shown only for the first three states.

The filter tracks the scan operations and the scan data at the TAP. As long as the sequence matches either α or β, the update operations are allowed. Otherwise, the trap state is reached, in which no further reconfiguration of the network is possible, and hence the protected segment `S4` remains inaccessible.

The filter can be extended with a single loop transition to allow repeated accesses to `S2` without the need to reconfigure `S1`. This transition is dashed in Figure 10.

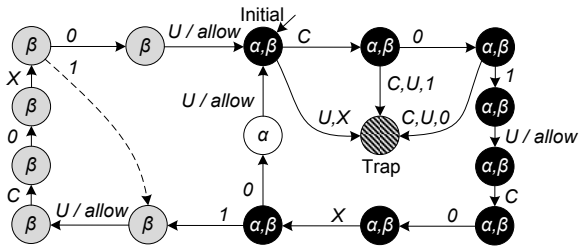


Fig. 10: The state diagram of a sequence filter allowing two sequences: α: *C01UC0X0U*, and β: *C01UC0X1UC0X0U*

7 Evaluation

The cost of the proposed protection method is evaluated on RSN architectures based on ITC'02 benchmarks [23]. Optimal (shortest) restricted access patterns are generated with the approach presented in [5] extended with the required constraints, as explained in Section 5. The sequence filters are constructed according to the algorithm from Section 6.1. The resulting state diagrams are transformed automatically into Verilog hardware models and synthesized for the Nangate 45 nm open cell library[1] with area optimization goal.

The area overhead is calculated w.r.t. the area of the scan network, which includes no system logic. The actual hardware overhead w.r.t. the full chip area is 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).

Restricted accesses are generated for random samples of target scan segments. Except for scan segments that configure the active scan path (*configuration scan segments*), all remaining scan segments are considered protected. The results discussed in the following sections, including the area overhead and the number of FSM states, represent average values acquired from the evaluation of 10 filters built for different random samples of target segments. The standard deviation of the area overhead is below 3% in the experiments.

7.1 Benchmark Circuits

Our approach is evaluated on two RSN architectures from [4]: 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 scan segment and a scan multiplexer that either bypasses or connects the lower-level scan segment (or a scan network hierarchy) to the higher-level scan chain, depending on the content of the configuration segment [15].

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 segments or sub-modules. For more details please refer to [4].

Table 1 describes the properties of the benchmark RSNs. For MUX-based architectures, the number of multiplexers is given in the second column, the total number of scan segments (including configuration segments) in the third column, the total number of scan cells (bits) in the fourth column, and the area for the Nangate 45 nm library in the fifth column. The characteristics of the SIB-based architectures are listed in the last four columns of Table 1.

7.2 Individual Segment Accesses

For each benchmark RSN, sequence filters are constructed for 10, 20, and 100 restricted accesses patterns. Each pattern realizes the shortest access to a single target scan segment, and the remaining segments are considered protected. This is relevant for low-latency access to individual segments.

As shown in Figure 11, the filter size depends on the number of allowed accesses: The area overhead ranges from 0.2 to 2.7% for 10 individual accesses. For 20 accesses, the area is 0.3 to 4.3%, and for 100 accesses it rises up to 10.6%. In most cases, the increase in area overhead is less than the increase in the number of allowed accesses. Note that twelve of the RSNs include about a hundred or less scan segments (cf. Table 1). For f2126, q12710, and a586710, even if individual access to a high fraction or all of their scan segments is allowed, the area overhead is below 1.7% .

The size of a sequence filter is proportional to the number of states in the filter's state diagram. State merging and sequence collapsing (cf. Section 6.2) considerably reduce the area overhead. Figure 12 shows the cumulative length of 100 restricted access patterns ("sequence bits") and the corresponding number of filter's states after state merging and sequence collapsing ("FSM states"). These techniques reduce the size of the state diagram by a factor of 2 at least, and by over 2 orders of magnitude for two benchmarks: q12710 and a586710.

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

Table 1: Characteristics of the benchmark scan networks

	MUX-based Architecture				SIB-based Architecture			
Design	#MUXes	#scan segments	#scan cells	Area [μm^2]	#SIBs	#scan segments	#scan cells	Area [μm^2]
d281	67	117	3 880	58 979	59	109	3 872	58 747
d695	178	335	8 407	127 007	168	325	8 397	126 777
h953	63	109	5 649	85 349	55	101	5 641	85 133
g1023	94	159	5 400	81 727	80	145	5 386	81 396
f2126	45	81	15 834	240 021	41	77	15 830	239 902
q12710	30	51	26 188	397 592	25	47	26 183	397 483
p22810	311	565	30 139	454 107	283	537	30 111	453 537
p34392	142	245	23 261	352 699	123	226	23 242	352 290
p93791	653	1241	98 637	1 486 772	621	1209	98 605	1 486 289
t512505	191	319	77 037	1 168 310	160	288	77 006	1 167 569
a586710	47	79	41 682	634 258	40	72	41 675	634 087

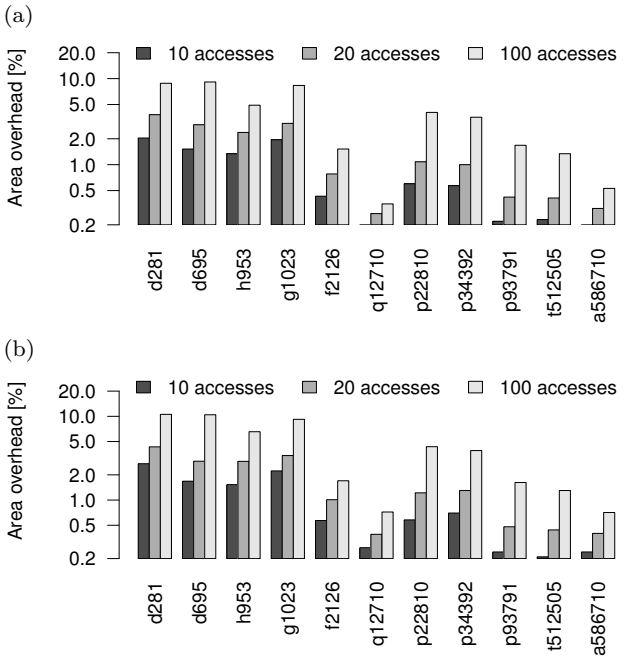


Fig. 11: Area overhead of sequence filters w.r.t. RSN area for (a) SIB-based and (b) MUX-based scan architecture

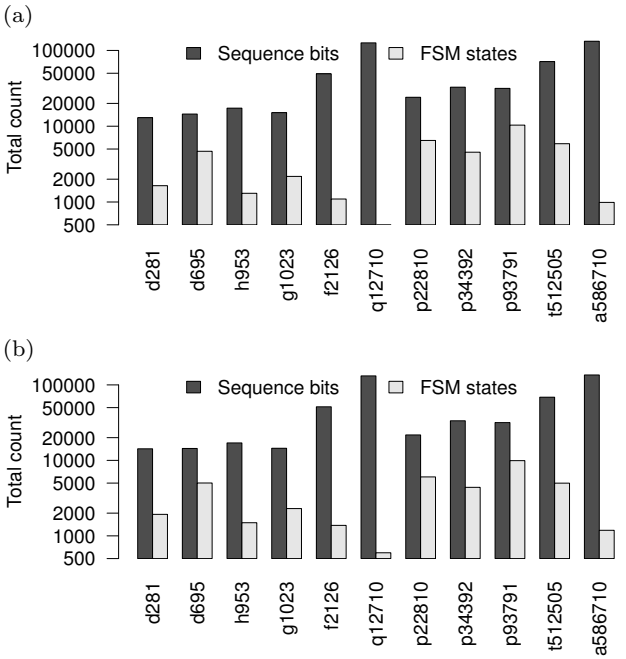


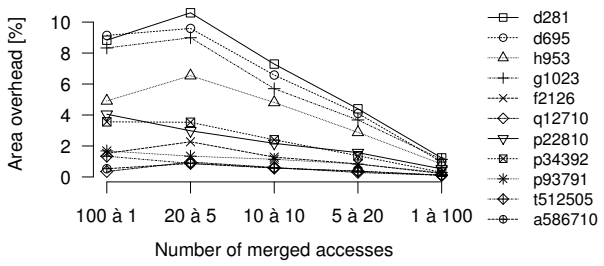
Fig. 12: Comparison of the total sequence length (in bits) and the number of FSM states after state merging and sequence collapsing for 100 restricted access patterns in (a) SIB-based and (b) MUX-based scan architecture

7.3 Concurrent Segment Accesses

In the second series of experiments, sequence filters are constructed for the concurrent access to 100 random scan segments realized by 1, 5, 10 and 20 restricted access patterns. The concurrent access is efficient if the target segments are usually accessed together.

Figure 13 shows area overhead of the resulting filters. For 20 accesses à 5 segments ("20 à 5"), area overhead of the resulting filters is close to the area for individual accesses ("100 à 1"). However, if the access to all 100 segments is realized with a *single* access pattern ("1 à 100"), the cost is reduced by a factor of 3 to 16 compared with the cost of individual accesses. Thus, if the segments are often accessed together, concurrent access has two benefits: The access times are lower, and the resulting sequence filters are smaller.

(a)



(b)

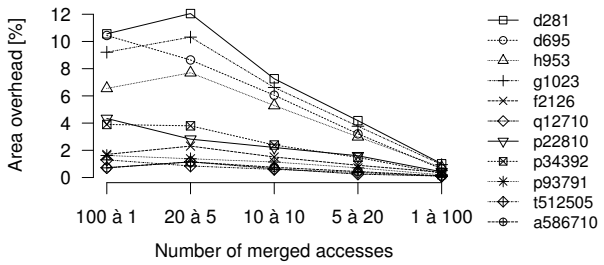


Fig. 13: Reduction of sequence filter overhead by merging the access to 100 scan segments in (a) SIB-based, and (b) MUX-based scan architecture

8 Conclusion

The accessibility offered by reconfigurable scan networks contradicts security and safety requirements for embedded instrumentation. Since such networks have distributed configuration and integrate a high number of instruments, state-of-the-art techniques for scan access protection are either ineffective or offer only coarse-grained security control. This paper presents a novel access protection method which requires only a *local* extension of the network's interface. The protected access port allows only a user-defined set of access patterns and prevents the access to protected instrumentation. This approach provides scalable fine-grained access management with low area overhead and can be combined with existing fuse- and authorization-based protection schemes.

References

1. Abramovici M (2008) In-System Silicon Validation and Debug. IEEE Design & Test of Computers 25(3):216–223
2. Agarwal K (2011) Secure Scan Design. US Patent App. 7,966,535
3. Baranowski R (2014) Reconfigurable Scan Networks: Formal Verification, Access Optimization, and Protection. PhD thesis, University of Stuttgart, URL http://elib.uni-stuttgart.de/opus/volltexte/2014/8982
4. Baranowski R, Kochte MA, Wunderlich HJ (2012) Modeling, Verification and Pattern Generation for Reconfigurable Scan Networks. In: Proc. IEEE International Test Conference (ITC), paper 8.2
5. Baranowski R, Kochte MA, Wunderlich HJ (2013) Scan Pattern Retargeting and Merging with Reduced Access Time. In: Proc. IEEE European Test Symposium (ETS), pp 39–45
6. Baranowski R, Kochte MA, Wunderlich HJ (2013) Securing Access to Reconfigurable Scan Networks. In: Proc. IEEE Asian Test Symposium (ATS), pp 295–300
7. Buskey R, Frosik B (2006) Protected JTAG. In: Proc. IEEE International Conference on Parallel Processing Workshops (ICCPW), pp 405–414
8. Chiu GM, Li JM (2012) A Secure Test Wrapper Design Against Internal and Boundary Scan Attacks for Embedded Cores. IEEE Trans on Very Large Scale Integration (VLSI) Systems 20(1):126–134
9. Clark C (2010) Anti-Tamper JTAG TAP Design Enables DRM to JTAG Registers and P1687 On-Chip Instruments. In: Proc. IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), pp 19–24
10. Da Rolt J, Das A, Di Natale G, Flottes ML, Rouzeyre B, Verbauwhede I (2014) Test versus Security: Past and Present. to appear in: IEEE Trans on Emerging Topics in Computing
11. Das A, Rolt J, Ghosh S, Seys S, Dupuis S, Natale G, Flottes ML, Rouzeyre B, Verbauwhede I (2013) Secure JTAG Implementation Using Schnorr Protocol. Journal of Electronic Testing (JETTA) 29(2):193–209
12. Dworak J, Crouch A, Potter J, Zygmontowicz A, Thornton M (2013) Don't Forget to Lock your SIB: Hiding Instruments using P1687. In: Proc. IEEE International Test Conference (ITC), paper 6.2
13. Ebrard E, Allard B, Candelier P, Waltz P (2009) Review of Fuse and Antifuse Solutions for Advanced Standard CMOS Technologies. Microelectronics Journal 40(12):1755–1765
14. Eklow B, Bennetts B (2006) New Techniques for Accessing Embedded Instrumentation: IEEE P1687 (IJTAG). In: Proc. IEEE European Test Symposium (ETS), pp 253–254
15. Ghani Zadegan F, Ingelsson U, Carlsson G, Larsson E (2012) Access Time Analysis for IEEE P1687. IEEE Trans on Computers 61(10):1459–1472

16. Hely D, Flottes ML, Bancel F, Rouzeyre B, Berard N, Renovell M (2004) Scan Design and Secure Chip [Secure IC Testing]. In: Proc. IEEE On-Line Testing Symposium (IOLTS), pp 219–224
17. IEEE (2013) IEEE Standard for Test Access Port and Boundary-Scan Architecture 1149.1-2013. Test Technology Technical Committee of the IEEE Computer Society, USA
18. Kömmerling O, Kuhn MG (1999) Design Principles for Tamper-Resistant Smartcard Processors. In: Proc. USENIX Workshop on Smartcard Technology (WOST), USENIX Association, pp 9–20
19. Larsson E, Ghani Zadegan F (2012) Accessing Embedded DfT Instruments with IEEE P1687. In: Proc. IEEE Asian Test Symposium (ATS), pp 71–76
20. Lee J, Tehranipoor M, Plusquellic J (2006) A Low-Cost Solution for Protecting IPs Against Scan-Based Side-Channel Attacks. In: Proc. IEEE VLSI Test Symposium (VTS), pp 94–99
21. Lee J, Tehranipoor M, Patel C, Plusquellic J (2007) Securing Designs against Scan-Based Side-Channel Attacks. IEEE Trans on Dependable and Secure Computing 4(4):325–336
22. Ley A (2009) Doing More with Less—An IEEE 1149.7 Embedded Tutorial: Standard for Reduced-Pin and Enhanced-Functionality Test Access Port and Boundary-Scan Architecture. In: Proc. IEEE International Test Conference (ITC), paper ET3.1
23. Marinissen E, Iyengar V, Chakrabarty K (2002) A Set of Benchmarks for Modular Testing of SOCs. In: Proc. IEEE International Test Conference (ITC), pp 519–528
24. Nicolaidis M, Noraz S, Courtois B (1989) A Generalized Theory of Fail-Safe Systems. In: International Symposium on Fault-Tolerant Computing (FTCS), Digest of Papers, pp 398–406
25. Park K, Yoo S, Kim T, Kim J (2010) JTAG Security System Based on Credentials. Journal of Electronic Testing (JETTA) 26:549–557
26. Park KY, Yoo SG, Kim J (2012) Debug Port Protection Mechanism for Secure Embedded Devices. IEEE Journal of Semiconductor Technology and Science 12(2):240–253
27. Pierce L, Tragoudas S (2013) Enhanced Secure Architecture for Joint Action Test Group Systems. IEEE Trans on Very Large Scale Integration (VLSI) Systems 21(7):1342–1345
28. Rearick J, Volz A (2006) A Case Study of Using IEEE P1687 (IJTAG) for High-Speed Serial I/O Characterization and Testing. In: Proc. IEEE International Test Conference (ITC), paper 10.2
29. Rearick J, Eklow B, Posse K, Crouch A, Bennetts B (2005) IJTAG (Internal JTAG): A Step Toward a DFT Standard. In: Proc. IEEE International Test Conference (ITC), paper 32.4
30. Rosenfeld K, Karri R (2010) Attacks and Defenses for JTAG. IEEE Design & Test of Computers 27(1):36–47
31. Rosenfeld K, Karri R (2011) Security-Aware SoC Test Access Mechanisms. In: Proc. IEEE VLSI Test Symposium (VTS), pp 100–104
32. Sourgen L (1992) Security Locks for Integrated Circuit. US Patent App. 5101121 A
33. Stollon N (2011) On-Chip Instrumentation: Design and Debug for Systems on Chip. Springer US
34. Tehranipoor M, Wang C (2011) Introduction to Hardware Security and Trust. Springer
35. Yang B, Wu K, Karri R (2004) Scan Based Side Channel Attack on Dedicated Hardware Implementations of Data Encryption Standard. In: Proc. IEEE International Test Conference (ITC), pp 339–344