Fine-Grained Access Management in Reconfigurable Scan Networks

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

IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD) Vol. 34(6) June 2015

doi: http://dx.doi.org/10.1109/TCAD.2015.2391266

Abstract: Modern VLSI designs incorporate a high amount of instrumentation that supports post-silicon validation and debug, volume test and diagnosis, as well as in-field system monitoring and maintenance. Reconfigurable scan architectures, as allowed by the novel IEEE Std 1149.1-2013 (JTAG) and IEEE Std 1687- 2014 (IJTAG), emerge as a scalable mechanism for access to such on-chip instruments. While the on-chip instrumentation is crucial for meeting quality, dependability, and time-to-market goals, it is prone to abuse and threatens system safety and security. A secure access management method is mandatory to assure that critical instruments be accessible to authorized entities only. This work presents a novel protection method for fine-grained access management in complex reconfigurable scan networks based on a challenge-response authentication protocol. The target scan network is extended with an authorization instrument and Secure Segment Insertion Bits (S^{2}IB) that together control the accessibility of individual instruments. To the best of the authors' knowledge, this is the first fine-grained access management scheme that scales well with the number of protected instruments and offers a high level of security. Compared with recent stateof- the-art techniques, this scheme is more favorable with respect to implementation cost, performance overhead, and provided security level.

Fine-Grained Access Management in Reconfigurable Scan Networks

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

ITI, University of Stuttgart, Pfaffenwaldring 47, D-70569, Stuttgart, Germany
Email: {baranowski, kochte, wu}@informatik.uni-stuttgart.de

***Abstract*—Modern VLSI designs incorporate a high amount of instrumentation that supports post-silicon validation and debug, volume test and diagnosis, as well as in-field system monitoring and maintenance. Reconfigurable scan architectures, as allowed by the novel IEEE Std 1149.1-2013 (JTAG) and IEEE Std 1687-2014 (IJTAG), emerge as a scalable mechanism for access to such on-chip instruments.**

While the on-chip instrumentation is crucial for meeting quality, dependability, and time-to-market goals, it is prone to abuse and threatens system safety and security. A secure access management method is mandatory to assure that critical instruments be accessible to authorized entities only.

This work presents a novel protection method for fine-grained access management in complex reconfigurable scan networks based on a challenge-response authentication protocol. The target scan network is extended with an *authorization instrument* and *Secure Segment Insertion Bits* (S^{2}IB) that together control the accessibility of individual instruments. To the best of the authors' knowledge, this is the first fine-grained access management scheme that scales well with the number of protected instruments and offers a high level of security. Compared with recent state-of-the-art techniques, this scheme is more favorable with respect to implementation cost, performance overhead, and provided security level.

***Index Terms*—Debug and diagnosis, reconfigurable scan network, IJTAG, IEEE Std 1687, secure DFT, hardware security, instrument protection**

I. Introduction

The rapidly increasing complexity and density of integrated circuits necessitates the use of various on-chip instruments to meet the targets for chip quality, time-to-market, dependability, and maintainability. Today's VLSI designs incorporate instrumentation for post-silicon validation and debug, volume test and diagnosis, as well as in-field system maintenance. Examples of on-chip instruments include embedded logic analyzers, trace buffers, test and debug controllers, assertion checkers, and physical sensors, to name just a few. Since the amount of embedded instrumentation in system-on-a-chip designs increases at an exponential rate, scalable mechanisms for instrument access and protection become indispensable [1], [2], [3], [4].

To reduce the interfacing cost, the embedded instruments are mainly accessed using scan techniques and are often interfaced as *data registers* with the widely adopted Test Access Port (TAP), as defined in IEEE Std 1149.1 (a.k.a. JTAG, [5]). As the number of instruments grows, Reconfigurable Scan Networks (RSNs) emerge as a scalable and cost-effective replacement for static 1149.1 data registers [6], [3]. In an RSN, the path through which data are shifted is configured by the state of *configuration registers* that can be arbitrarily distributed over the RSN. The IEEE Std 1687 (or IJTAG for *Internal* JTAG) standardizes the design and access to this type of scan networks. It allows flexible architectures with distributed and hierarchical configuration for efficient access to on-chip instrumentation [3], [4].

The most popular type of RSN architectures uses gateways called *Segment Insertion Bits* (SIB) for configurable access to on-chip instrumentation [3], [7]. This is a simple and flexible architecture compliant with IJTAG that allows hierarchical control over the accessibility of individual instruments. A similar bypassing mechanism is also proposed in the novel IEEE Std 1149.1-2013 in form of *segment selectors* and *excludable segments* [5].

An example of a three-level SIB hierarchy is shown in Figure 1. A SIB is in principle a configurable bypass: It either bypasses a subordinate instrument or sub-network connected to its TO/FROM ports, or connects it to the higher-level scan chain (between SI and SO ports). The mode of operation is chosen by shifting a single configuration bit into the SIB's SI port.

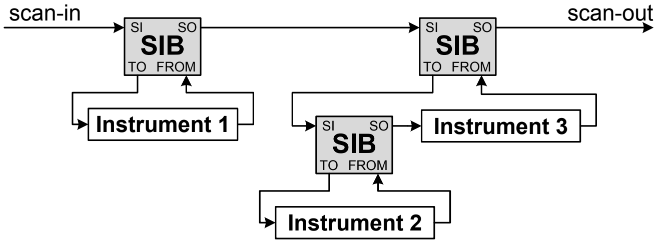


Fig. 1. Example of a Reconfigurable Scan Network (RSN) based on Segment Insertion Bits (SIB)

The improved accessibility of on-chip instrumentation may contradict security and safety requirements for chip internals [8]. Embedded instruments and the scan infrastructure can be abused for sabotage, unlicensed usage, or *Intellectual Property* (IP) theft. An attacker may exploit the scan infrastructure to gain access to protected data (password or IP), alter the system state by fault injection, or perform illegal operations. Successful attacks on the 1149.1 TAP are reported for pirating satellite TV services, circumventing mechanisms for Data Rights Management (DRM) [9], unlocking protected services of mobile phones [10], or retrieval of keys from cryptographic cores [11].

Different levels of infrastructure accessibility are required during product development, volume production, and in-field operation. 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 to prevent e.g. IP theft or tampering. Moreover, different accessibility levels may be required depending on the eligibility of the accessing entity. In automotive applications, for instance, full accessibility is mandatory during manufacturing and assembly test, while only limited access is allowed during maintenance in a workshop to prevent unauthorized chip tuning.

The goal of this work is to provide a secure and cost-efficient mechanism for access management in complex reconfigurable scan networks compliant with IEEE Std 1687, with a particular focus on the popular SIB-based architecture. This mechanism shall assure that only authorized entities are allowed to access protected instruments. Moreover, the protection shall allow for distinct permissions for multiple authorized entities.

We reach this goal with an *authorization instrument* that is integrated into the target RSN and realizes a challenge-response authentication protocol. Individual instruments are protected by so called *Secure Segment Insertion Bits* (S^{2}IB). This approach integrates well with hierarchical SIB-based designs and requires only a minor modification of the original design. Since a S^{2}IB is only slightly larger than a regular SIB, our protection scheme scales well with the number of protected instruments. As S^{2}IBs are daisy-chained, no additional global signals for security control are required. Furthermore, except for the initial constant overhead of unlocking protected instruments, the S^{2}IB-based protection does not increase the access time, and the original scan data (scan-in bit sequences) for accessing instruments in the original RSN can be directly reused to access the protected design. Experimental results show that the one-time authentication overhead is only 788 clock cycles for unlocking any combination of 256 protected instruments. If basic cryptographic primitives are already available on the chip and can be reused, the area overhead of the proposed approach is marginal.

The next section formulates the access management problem formally, discusses state-of-the-art techniques for scan network protection, and compares them with the proposed approach. Section III provides the technical background and terminology of reconfigurable scan networks. The proposed access management scheme is detailed in Section IV, followed with an analysis of its security. The area overhead is evaluated in Section V.

II. Problem Formulation and Related Work

The problem is formulated formally as follows. Given is:

- a set of on-chip instruments $I = \{i_1, i_2, \ldots, i_n\}$ and a subset of *protected* instruments $I_P \subseteq I$ (e.g. scan chains; debug, test, reprogramming, or reconfiguration controllers),
- a set of authorized entities $E = \{e_1, e_2, \ldots, e_m\}$ (e.g. manufacturing contractors, next-tier companies in the supply chain, service personnel, software/hardware tools of end users),
- access *permissions* in form of a mapping from protected instruments to subsets of authorized entities: $p : I_P \to \mathcal{P}(E)$.

The goal is to protect the interface of the scan infrastructure so that any protected instrument $i \in I_P$ can only be accessed by the subset of authorized entities $p(i)$, and all unprotected instruments in $I \setminus I_P$ remain accessible to everybody.

Partial solutions to this problem have been proposed in the past. The most prominent techniques are discussed below. For a more complete review of state-of-the-art protection techniques for simple scan infrastructures please refer to the recent survey in [8].

If all on-chip instruments are protected ($I = I_P$) and no authorized entity exists ($E = \varnothing$), the physical interface of the scan infrastructure (e.g. 1149.1 TAP) can be completely disabled. This is usually done after manufacturing test, e.g. using One Time Programmable (OTP) memory cells called *fuses* [12], or by physical TAP removal using a wafer saw [13]. 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.

If only a subset of instruments is protected ($I_p \subset I$) and no authorized entity exists ($E = \varnothing$), either the protected instruments themselves or some instructions of the 1149.1 TAP controller can be permanently disabled using on-chip fuses [14]. 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 [9].

To manage distinct access rights of different entities ($E \neq \varnothing$), an *entity authentication* mechanism is required. In simplest schemes, each entity $e \in E$ is assigned a *secret* k_e (a string of bits of arbitrary length), the possession of which must be proven to the chip to unlock respective instruments $\{ \ i \in I_P \mid e \in p(i) \ \}$. Alternatively, each protected instrument $i \in I_p$ can be associated with a *secret* k_i that is known only to all authorized entities $p(i)$ or only to a *secure server* that can serve the requests of these entities.

Weak authentication schemes involve a static secret (*password*) that is presented to the chip [15], [16], [17], [18], [19], [20]: To access a protected instrument $i \in I_P$, the secret k_i is directly applied to dedicated primary inputs [15], embedded at constant [16], [18] or variable [20] positions in scan data (scan-in bit sequence), or written to a dedicated data register in a 1149.1 circuitry [17] or an IEEE Std 1500 wrapper [19]. 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 security requirements.

Stronger authentication schemes, which do not reveal the secret during communication with the chip, are based on challenge-response protocols [21], [22], [23], [24], [25], [26], [27]: In these schemes, upon a request of an entity $e \in E$

to access a protected instrument $i \in I_P$, the chip provides a non-repeating *challenge* value and expects the entity to provide an expected *response* value. The response is calculated from the challenge, the secret k_i, and possibly the secret k_e, using a specified cryptographic algorithm, e.g. a one-way hash function, symmetric- or public-key cryptography [28]. The secret k_i may be directly available to the entity [22], or may only be available to a *secure server* that is assigned the task of authentifying the entity e and providing it online with a response to the challenge if $e \in p(i)$ [21]. To reduce the amount of online communication, the secure server can also issue *credentials* which can later be used by the entity to gain offline access to the chip [24]. To limit the number of allowed offline accesses, a similar approach based on *authentication tokens* was later proposed by the same authors [25]. For provably secure mutual authentication, an application of the Schnorr protocol was proposed recently in [26].

Alternatively, scan data encryption and Message Authentication Codes (MAC) can be used to implement basic access rights management, as proposed in [23]. To manage different permissions for different instruments, the encryption circuitry is distributed over the chip to locally decrypt scan input and encrypt scan output data [29]. This technique has the benefit that scan data are never exposed in plaintext, not even to other on-chip components through which these data are shifted.

The above-mentioned access management mechanisms based on authentication and encryption are reasonably secure and allow for distinct access rights for different entities or users. However, while it is desired that the access rights are managed at the level of individual on-chip instruments distributed over the system-wide scan infrastructure, the techniques presented in [21], [24], [25], [26] allow access management at the level of 1149.1 TAP instructions only. Fine-grained access management for individual instruments can be achieved by connecting *each* protected instrument with the authorization controller, as proposed in [22], [27]. However, the scalability of this approach is limited due to high routing overhead. Likewise, the local scan data encryption [29] becomes unwieldy and incurs high hardware cost if many instruments need protection.

Recently, we proposed a method for protecting individual instruments using *sequence filters* that are placed locally at the TAP and require no additional global wiring and no modification of the infrastructure [30], [31]. A sequence filter monitors all access operations and blocks them if they do not follow an allowed pattern. The filter can be activated statically with an on-chip fuse or deactivated for authorized entities using any authentication mechanism. While this approach allows for fine-grained access management in arbitrary reconfigurable scan networks, the hardware overhead of the filters may be high if the majority of instruments is *not* protected.

To the best knowledge of the authors, the only existing fine-grained access management technique tailored for SIB-based RSNs has been recently proposed in [20]. In this technique, the SIBs that enclose protected instruments are replaced with so called *Locking Segment Insertion Bits* (LSIB). An LSIB is open only after a predefined multi-bit *key* is loaded into a shift register that may be distributed over the entire RSN. This way, each instrument can be protected individually, but the unlocking time (access overhead) is proportional to the number of protected instruments. If dedicated shift registers are introduced for the keys, each LSIB entails the hardware overhead of dozens of sequential elements. This overhead is reduced if existing scan chains can be reused for this purpose. In the latter case, however, each LSIB requires the routing of dozens of possibly long wires and needs careful design of the access mechanism to minimize access overhead. Therefore, this approach is most practical if only a few instruments need individual protection. To prevent that the keys are exposed while communicating with the chip, additional scan data obfuscation techniques are required, as in [32].

The work at hand presents the first scalable access authorization mechanism based on challenge-response authentication that can be easily integrated into any complex RSN design. Our technique bears the following characteristics:

- It features fine-grained control over the access to individual instruments. Each authorized entity may be assigned distinct access rights for each protected instrument. The number of individually protected instruments and distinct authorized entities is unlimited.
- It is easily portable as the authorization instrument can be integrated into any hierarchy level of an existing RSN or using a dedicated RSN. In 3D integrated circuits, each die can be equipped with an authorization instrument controlling the access to protected instruments on the die.
- Secrets are not exposed while communicating with the chip and need not be distributed to authorized entities.
- Although the instruments are protected individually, just a single constant-time access is needed to unlock any combination of them. The protection causes minimal or no access overhead during regular access to the RSN after authentication.
- Since the protected instruments are accessed via a scan chain, the proposed approach causes no routing issues.
- Experimental results show that the area overhead is only slightly sensitive to the number of protected instruments, and it is negligible if a hash core and a random number generator are already available on-chip.

If slight modifications of the scan infrastructure are allowed, the authorization mechanism presented in this paper is preferable to our previous technique based on sequence filters [30], [31], especially when just a small fraction of instruments is protected. If required, the security of the proposed scheme can be further improved by replacing the underlying challenge-response protocol with other authentication mechanisms, e.g. based on credentials or authentication tokens, as in [25], or by using the Schnorr protocol, as in [26].

III. Reconfigurable Scan Networks

This paper deals with the protection of reconfigurable scan networks (RSNs) as defined in the novel IEEE Std 1687-2014 and 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 [4].

RSNs are usually accessed through a 1149.1-compliant Test Access Port (TAP) [5] 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 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, an optional *shadow* register is loaded with data from the shift register. The shadow register is stable during the shift operation (as in 1149.1 test data registers). A scan segment with a shadow register may be used for bidirectional communication with an on-chip instrument. 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.

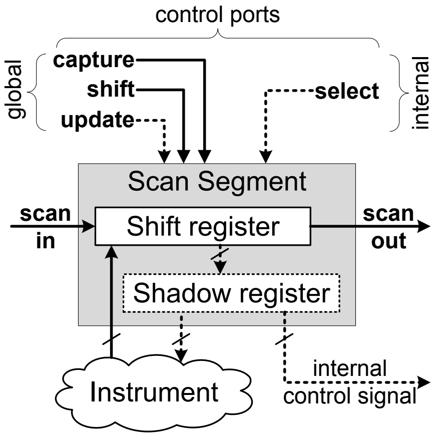


Fig. 2. Scan segment

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 or external control inputs.

An RSN has a *primary scan-input* and a *primary scan-output*, a *reset* input, 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 1149.1 TAP, these signals are driven by the TAP controller.

A *scan path* is a non-circular sequence of daisy-chained scan segments starting at a primary scan-in port and ending at a 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 basic access to a scan network consists of three phases, as defined by IEEE Std 1149.1 [5]: *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 register in the network requires that the accessed register is part of an active scan path. A *scan access* is a sequence of CSU operations required to reconfigure the scan network and access the target registers.

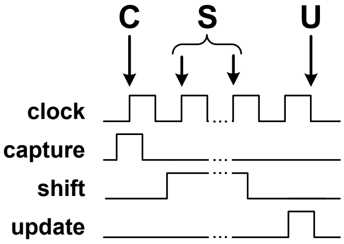


Fig. 3. Capture, Shift, Update (CSU) operation

Hierarchical RSN architectures based on Segment Insertion Bits (SIB, cf. Figure 1) have recently gained attention due to their simplicity, regularity, as well as fair flexibility and performance. A possible implementation of a SIB is presented in Figure 4: It contains a 1-bit shift register "S" and a 1-bit shadow register "U" (D-type flip-flops) that are controlled by the global control signals and the *select* signal in the same way as the registers of a scan segment (cf. Figure 2). In particular, register U is only loaded from S upon an *update* operation when the *select* signal is active. The content of U constitutes the SIB *configuration*: the lower level segment is connected between the Scan Input (SI) and Scan Output (SO) ports when U=1 and *select*=1, i.e. when *to-select*=1; otherwise it is bypassed. The scan path through a SIB always contains the shift register S. Different SIB implementations and their properties are discussed in [7].

IV. Access Authorization Mechanism

In the following, we present the novel access authorization mechanism based on an authorization instrument and Secure SIBs (S^{2}IB). We start with a short overview of the proposed method, followed by a detailed description of the authorization instrument and the S^{2}IBs. Finally, we discuss the challenge-response authentication protocol and analyze the security of this approach.

A. Authorization Principles

In the proposed approach, each protected instrument $i \in I_P$ is associated with a unique secret k_i that is stored on-chip. To unlock a set of instruments $a, b, \ldots, c \subseteq I_P$, a requesting

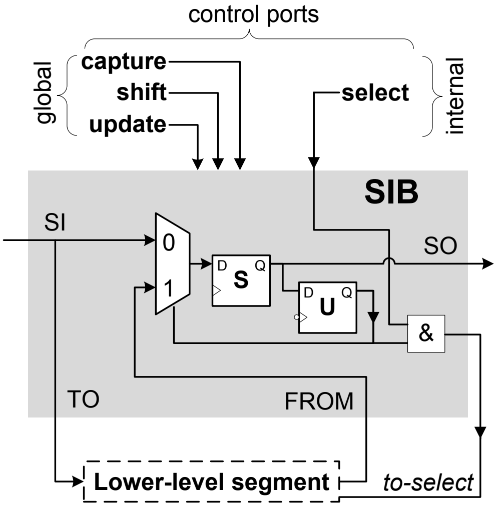


Fig. 4. Example implementation of a Segment Insertion Bit (SIB)

entity e must prove to the chip that it knows all respective secrets, i.e. $k_a, k_b, \ldots, k_c$. To reduce the chance that the secrets are revealed, the authentication process follows a challenge-response protocol, as described below.

Initially, all protected instruments I_P are locked. The chip takes requests from any entity e to unlock a specified set of protected instruments. Upon such a request, the chip generates a challenge, i.e. a cryptographic *nonce*. The expected response is calculated by applying a one-way hash function to the challenge followed with the secrets associated with the requested instruments, i.e.:

$$\text{response} = \text{hash}\left(\text{challenge} \mid k_a \mid k_b \mid \ldots \mid k_c\right).$$

(Note that the secrets $k_a, k_b, \ldots, k_c$ must be fed to the hash function in a well defined order.) The protected instruments are unlocked only if the response presented by e matches the expected response calculated on-chip. The requesting entity e may either be in possession of the necessary secrets and calculate the response itself, or it may delegate the challenge to a *secure server* that authentifies e and provides it online with the correct response if $\forall_{i \in (a,b,\ldots,c)}\ e \in p(i)$.

The protection is realized by extending the RSN with an *authorization instrument* that is accessed by the user to perform authentication. Additionally, each SIB that directly encloses a protected instrument in the original RSN is replaced with a *Secure Segment Insertion Bit* (S^{2}IB). (Alternatively, each sub-network of protected instruments sharing the same secret can be enclosed within a single S^{2}IB.) All S^{2}IBs within the RSN are daisy-chained and form a *Secured Scan Chain* (SSC) that is accessed to unlock or re-lock protected instruments. The authorization instrument includes an *authorization controller* that takes care of the challenge-response authentication process and controls the access to the SSC.

Figure 5 presents a protected version of the RSN from Figure 1, with protected instruments 1 and 2. The SIBs enclosing the protected instruments are replaced with S^{2}IBs and connected into a secure scan chain which crosses different hierarchy levels of the original RSN (SSC, represented with a dashed line in Figure 5). In this example, the authorization instrument is integrated into the top-level chain via SIB$_1$. To reduce access time overhead, this instrument can instead be integrated as a separate chain (separate data register of the 1149.1 TAP), or it can be embedded at deeper levels of RSN hierarchy. For the sake of readability, the distribution of local *select* signals and the global *SSC-select*, *capture*, *shift*, *update* and *secured-update* signals is omitted in Figure 5.

B. Authorization Instrument

The authorization instrument includes a scan segment called *interface* which is daisy-chained with SIB$_2$ that encloses the secured scan chain (SSC, cf. Figure 5). The *interface* is just a multi-bit shift register which is used for the communication of requests, challenges, and responses. It contains no shadow register and its length is chosen as the maximum of the challenge, response, and SSC lengths. Upon a *capture* operation, the *interface* segment is loaded with the value presented by the *challenge* output of the authorization controller. Its state is visible to the authorization controller over the *response* input.

The authorization controller is responsible for assuring that no unauthorized entity can unlock protected instruments by reconfiguring the SSC. It observes the *to-select* output of SIB$_1$ (via *auth-select*), the *to-select* output of SIB$_2$ (via *SSC-select*), and the scan data that are shifted into the SSC (via *SSC-sense*). In case of an unauthorized access to the SSC, the controller blocks the *update* operation for the entire RSN, which prevents unauthorized unlocking of the S^{2}IBs. This is realized by setting the *update* signal to 0 if any unauthorized access to the SSC is attempted. The resulting *secured-update* signal (cf. Figure 5) is routed to the RSN (including the authorization instrument itself) in place of the original *update* signal (controlled e.g. by the TAP controller), similar as in [30], [31], [27].

The authorization controller requires (cf. Figure 5):

- a hardware component for the generation of *nonces*, i.e. non-repeating challenges, to prevent *reply attacks*—in the simplest case a Random Number Generator (RNG) or preferably a True RNG (TRNG) [28],
- a cryptographic hash core for the calculation of responses,
- a read-only memory holding the secret for each instrument (*secret memory*).

Just a single RNG and hash core is required, regardless of how many instruments are protected. If such components are already available on-chip and are accessible through a secure system bus, they can be reused to minimize protection cost. The size of secrets, challenges, and responses (number of bits) can be tailored to meet a trade-off between security level and area overhead.

C. Secure Segment Insertion Bit (S^{2}IB)

The implementation of a S^{2}IB is presented in Figure 6. The S^{2}IB is an extension of the regular SIB (cf. Figure 4) with an additional shift register "S2" and its corresponding shadow register "U2". The S^{2}IB is *locked* (closed) if U2 is set to 0, otherwise it is *unlocked* (can be open). The protected

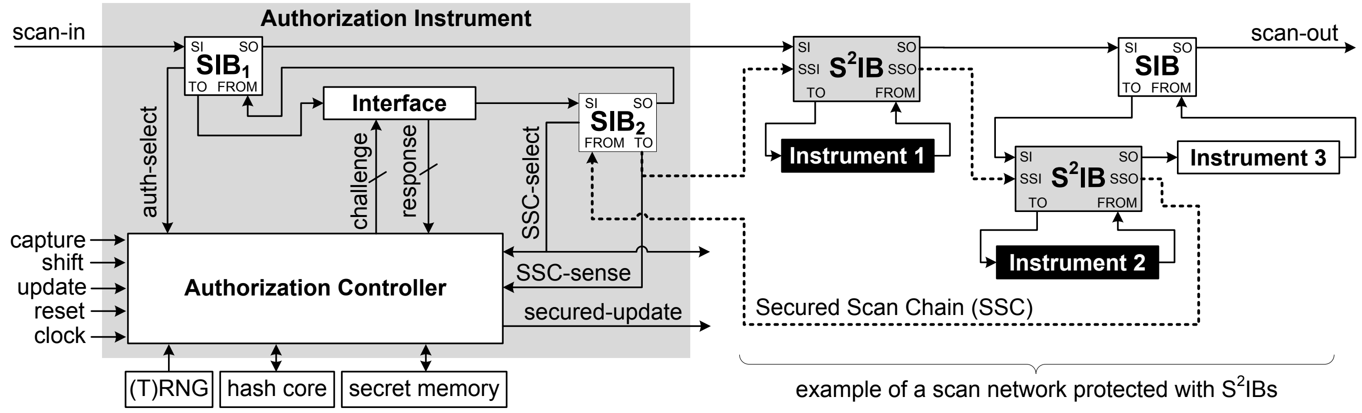


Fig. 5. Authorization instrument attached to a scan network with instruments 1 & 2 protected by Secure SIBs (S^2IB)

instrument (or lower-level scan network) is accessible only if both shadow registers, U1 and U2, are set to 1. Upon an *update* operation, U2 is loaded with the value stored in S2 only if the *SSC-select* port is active. The additional shift register S2 is accessed via the Secure Scan Input (SSI) and Output (SSO) ports when *SSC-select* port is active. SSI and SSO are used to form the SSC (cf. Figure 5). Upon a *capture* operation with an active *SSC-select* port and during a *reset* operation, S2 is loaded with value 0 to prevent unintentional unlocking.

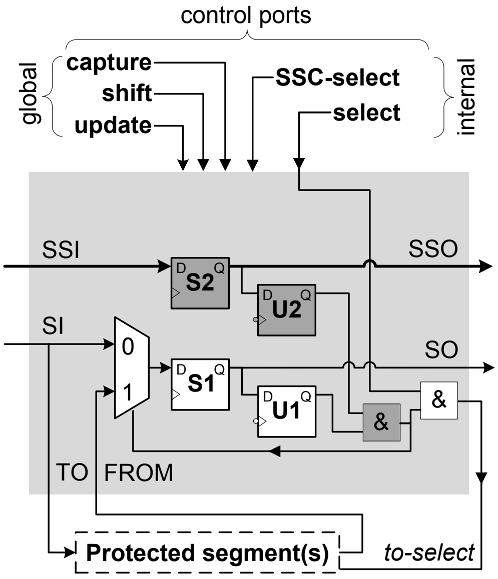


Fig. 6. Implementation of the Secure Segment Insertion Bit (S^2IB), extensions w.r.t. SIB are shown in dark gray

D. Challenge-Response Protocol

Below we explain step by step the challenge-response protocol from the perspective of a requesting entity, and we list the actions taken by the authorization controller. The authorization protocol is also illustrated in Figure 7. Each step corresponds to a single CSU operation that reads/writes the scan segments on the active scan path. We assume that initially SIB_1 and SIB_2 are closed (cf. Figure 5).

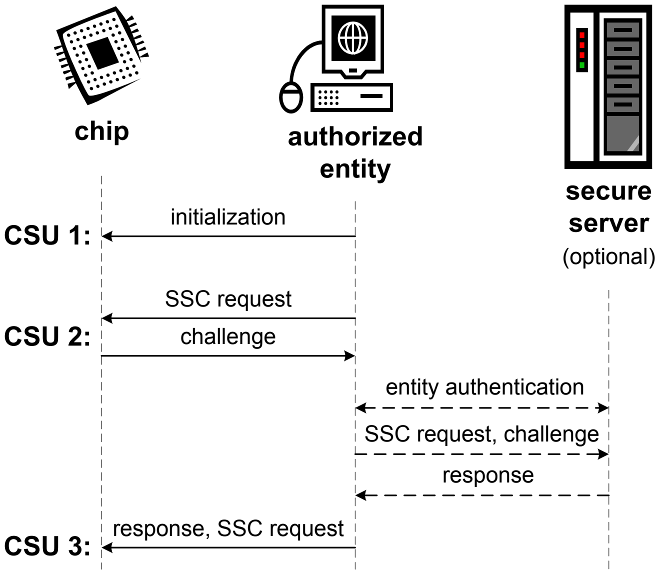


Fig. 7. Authorization protocol

1) Shift value 1 to SIB_1 (so that the *interface* segment becomes accessible in the next CSU).
   - The authorization controller remains in its reset state as long as SIB_1 is closed (signal *auth-select* is inactive).
   - In the reset state, the *secured-update* signal forwards the external *update*.
2) Shift a challenge value out of the *interface* segment. Shift the requested configuration of the SSC into the *interface* (with 1s for the S^2IBs that are to be unlocked and 0s for the remaining S^2IBs). Shift value 1 to SIB_1 and SIB_2.
   - The authorization controller generates and stores the challenge value.
   - The requested configuration of the SSC is also stored.
   - After the CSU operation completes, the authorization controller starts calculating the expected response: it hashes the stored challenge with secrets of each requested instrument, in the order of instruments in

the SSC. This calculation can be done while serving the subsequent CSU operation. The same calculation is expected to be done by the requesting entity or a secure server.
    - The *secured-update* signal forwards the external *update* as long as SIB_2 is closed (*SSC-select* is inactive).
3) Shift the response value into the *interface*. Shift the previously requested configuration into the SSC. Shift value 0 to SIB_1 and SIB_2.
    - The authorization controller observes the configuration shifted into the SSC via *SSC-sense* to ensure that it matches the requested configuration from the previous CSU operation.
    - As soon as SIB_2 becomes open (*SSC-select* is active), the following holds for the *secured-update* signal: Unless the configuration currently shifted into the SSC matches the requested configuration and the content of *interface* matches the response calculated by the controller, *secured-update* is set to 0 (reconfiguration is blocked). Otherwise it forwards the external *update* signal (reconfiguration is allowed).
4) Access the unlocked RSN normally, until the scan infrastructure is reset or until the S^2IBs are locked.
    - The authorization controller is back in its reset state.

E. Security Analysis

The proposed access management technique provides logical security: Assuming that the chip is operated under normal conditions (clock signal, temperature, electromagnetic interference, etc. are within specification), the proposed scheme assures that the protected instruments are only accessible to authorized users who know the respective secrets. Assuming that the cryptographic primitives (TRNG, hash core) are secure and invulnerable to side-channel attacks, the achieved security level depends primarily on the size of the challenges, responses, and secrets.

The presented method is orthogonal to protection techniques against invasive attacks (e.g. chip dismantling, reverse-engineering, microprobing) and non-invasive attacks (fault injection, side-channel analysis). Depending on the target security level, the scan infrastructure may still need to be protected against such attacks, e.g. using the methods reviewed in [9]. For instance, to provide protection against fault-injection attacks, the design must be equipped with sensors that can detect under/over voltage, extreme temperatures, as well as clock and reset instability or glitches. If any abnormality is detected, the *secured-update* signal must be set to 0.

We note that the proposed protection scheme assures secure authentication but not secure communication. To prevent that an adversary takes over the chip after some protected instruments have been unlocked by an authorized entity, the communication with the chip must be implemented over an *authentic channel*, i.e., a channel that an adversary may eavesdrop on, but must not be able to tamper with. To assure security, the authorized entity must either perform the access in a secure environment (the chip must not be physically accessible to any adversary), or must secure the channel, e.g. using Message Authentication Codes (MAC) as in [23].

V. Evaluation

In the following, we evaluate the performance and area overhead of the proposed protection scheme and compare it with the LSIB-based protection from [20].

A. Experimental Setup

We assume that the target application requires a very high security level with access management over individual protected instruments. The authorization mechanism is therefore implemented with the following parameters:

- 256-bit challenge and response,
- 128-bit secret for each protected instrument,
- support for up to 256 individually protected instruments (S^2IBs).

These parameters can be tailored so as to balance the security level with protection cost. Note that to accommodate more than 256 S^2IBs, just the *interface* register length and the capacity of the secret memory would need to be extended.

The authorization controller uses a hardware core to calculate the hash function. We evaluate three configurations of our protection scheme with different hash functions, using:

- SHA-1 core with a 160-bit digest (least secure),
- SHA-2 core with a 256-bit digest,
- SHA-3 (Keccak) core with a 512-bit digest (most secure).

The hash cores have been obtained from OpenCores.[1] In the last configuration, the 512-bit SHA-3 digest is truncated to obtain a 256-bit response.

Our protection scheme supports distinct access rights for authorized entities: Each instrument can be unlocked or locked individually with a unique secret. The read-only secret memory is implemented as a combinational logic block that takes the instrument number (address) and provides the key value. We synthesize the secret memory for random key values to assure that the keys cannot be derived from each other.

For a fair comparison, we require the same granularity of access management from the LSIB-based architecture, and hence we let each LSIB be enabled with a unique key. Furthermore, we assume that the scan segments which hold the LSIB keys cannot be shared with system logic and require dedicated hardware. This is justified by the fact that sharing may result in unpredictable routing issues, may require extensive redesign of the original RSN architecture, and is difficult to benchmark since it heavily depends on the target system architecture. Note that these assuptions result in worst-case area overhead for the LSIB approach.

Both the proposed protection scheme and the LSIB architecture are synthesized for the Nangate 45nm open cell library [33] with area optimization goal. Just to give the reader a point of reference, in Table I we report the properties of some SIB-based RSN benchmarks that were obtained from ITC'02 cores [34]. Figure 8 shows the SIB-based architecture for the top-level part of the p34392 benchmark. In these benchmarks,

[1] http://www.opencores.org

SIBs are used as hierarchical gateways to the modules, their submodules, input and output boundary registers, and internal scan chains. Column "# Levels" gives the number of hierarchy levels in the corresponding ITC'02 benchmark. The number of scan segments ("# Scan segments") includes the 1-bit scan segments that comprise the SIBs. The last column gives the total benchmark area in Nangate 45 nm. Please note that this area refers to the area of the RSN only, i.e., it does not include the system logic.

TABLE I: CHARACTERISTICS OF ITC'02 BENCHMARK CIRCUITS

Design	# Levels	# SIBs	# Scan segments	# Scan cells	Area $[\mu m^2]$
u226	2	50	90	1 466	22 313
d281	2	59	109	3 872	58 747
d695	2	168	325	8 397	126 777
h953	2	55	101	5 641	85 133
g1023	2	80	145	5 386	81 396
f2126	2	41	77	15 830	239 902
q12710	2	25	47	26 183	397 483
p22810	3	283	537	30 111	453 537
p34392	3	123	226	23 242	352 290
p93791	3	621	1 209	98 605	1 486 289
t512505	2	160	288	77 006	1 167 569
a586710	3	40	72	41 675	634 087

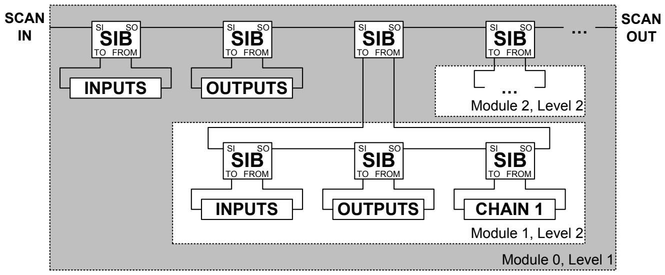


Fig. 8. SIB-based scan architecture for the p34392 benchmark

B. Performance Overhead

The authorization is conducted only once per test or maintenance session, and hence introduces only a one-time, constant access overhead. During the authorization process, the user can simultaneously perform other accesses to non-protected instruments. We assume that a CSU operation, apart from the shift cycles, requires 5 clock cycles for the update and capture phases, which is a typical overhead for an 1149.1-compliant TAP [5]. We further assume that the hash core offers sufficient throughput to compute the hash of the challenge and 256 secrets (for unlocking all instruments) while the third CSU operation is in progress (cf. Figure 7), which is the case in all our implementations. Under these assumptions, the authorization process takes:

- $5 + 1$ cycles for opening the authorization instrument in the first CSU operation,
- $5 + 256 + 2$ cycles for communicating the challenge and request in the second CSU operation,
- $5 + 256 + 2 + N$ cycles for communicating the response and request in the third CSU operation, where N is the total number of S^{2}IBs (SSC length).

This totals to $532 + N$ clock cycles to unlock or re-lock any combination of protected instruments.

During a regular RSN access after authentication, the authorization instrument causes an overhead of one shift cycle per CSU operation due to the closed SIB$_1$ (cf. Figure 5). To minimize or eliminate this effect, the authorization instrument can be placed at deeper hierarchy levels or connected to the 1149.1 TAP as a separate data register.

Since the proposed technique extends the design with just a single additional scan chain (SSC) and a single AND gate on the global *update* signal, it has minimal impact on the routing and the target operating frequency of the system. The proposed approach is therefore more favorable than [22], [27] which require that the authorization controller be wired with each protected instrument individually.

If more than just a few instruments need protection, our constant-time authorization scheme is also more preferable to the LSIB-based approach [20], which requires that a unique secret be shifted into the network for each protected instrument individually (preferably, each with a length of at least 48 bits). Due to the high amount of key registers, the LSIB-based architecture may also cause a higher performance penalty for the access to unprotected instruments.

C. Area Overhead

In the following analysis, we neglect the area required for the Random Number Generator (RNG) for two reasons: (1) If a (T)RNG core is already available on-chip, it can be reused. (2) In our scheme, random numbers need to be generated at a very low rate, which allows a very cost-efficient TRNG implementation (e.g. using ring oscillators, as in [22]).

Table II shows the area overhead of the proposed protection and the LSIB-based approach [20]. The first column gives the number of protected instruments. Columns 2-6 detail the area required by the proposed approach, i.e., the overhead of:

- Col. 2: authorization instrument (cf. Figure 5),
- Col. 3: secret memory with 128-bit keys,
- Col. 4-6: total protection overhead using SHA-1, SHA-2, and SHA-3 cores, respectively (without RNG).

The area overhead of the LSIB-based approach with 48-, 64-, and 128-bit keys is listed in columns 7-9, respectively.

If the hash core can be reused, the proposed access management scheme requires $3\,467\,\mu m^2$ for a single protected instrument, and only 4.7 times more if as much as 256 instruments need individual protection ($16\,171\,\mu m^2$). This way, for instance, almost all scan segments in the t512505 benchmark can be protected at only 1.4% area overhead with respect to the scan infrastructure alone (without system logic, cf. Table I). With the same secret length (128-bit key), the LSIB-based approach is slightly cheaper for a *single* protected instrument, but is already more costly if two instruments need protection. For 256 instruments, the S^{2}IB-based approach is cheaper by a factor of 31. The overhead of the LSIB-based approach is proportional to the key-length, but even with 48-bit keys the area overhead is prohibitive if many instruments need protection (cf. area of benchmark RSNs in Table I).

TABLE II: HARDWARE OVERHEAD OF THE PROPOSED S^{2}IB-BASED PROTECTION SCHEME AND THE RELATED WORK BASED ON LSIBS

	S^{2}IB-based protection					LSIB-based protection [20]		
# protected instruments	authorization instrument [μm^2]	secret memory [μm^2]	total area [μm^2]			total area [μm^2]		
			with SHA-1	with SHA-2	with SHA-3	48-bit keys	64-bit keys	128-bit keys
(1)	*(2)*	*(3)*	*(4)*	*(5)*	*(6)*	*(7)*	*(8)*	*(9)*
1	3 467	0	11 386	14 485	28 661	766	1 016	2 003
2	3 514	1	11 434	14 533	28 709	1 503	2 003	3 974
4	3 588	15	11 522	14 620	28 797	2 983	3 975	7 917
8	3 736	113	11 768	14 867	29 043	5 929	7 913	15 801
16	3 993	245	12 158	15 256	29 433	11 853	15 836	31 557
32	4 538	502	12 959	16 057	30 234	23 669	31 609	63 188
64	5 628	1 032	14 579	17 678	31 854	47 466	63 109	126 333
128	7 813	2 116	17 849	20 947	35 123	94 529	126 531	252 372
256	12 205	3 966	24 091	27 190	41 366	188 631	252 976	504 727

If no hash core can be reused, the area overhead of the S^{2}IB-based access management with a dedicated SHA-1 core is same as the area of the LSIB-based approach for about 6 protected instruments with 128-bit keys (col. 4 in Table II). With a SHA-2 core, this threshold is between 7 an 8 instruments, and with SHA-3, it is close to 15 instruments. For 256 protected instruments, the S^{2}IB-based protection with a SHA-1 core requires 8 times less area than 48-bit LSIBs, and 21 times less than 128-bit LSIBs. Due to the challenge-response protocol used by the S^{2}IB-based scheme, the implementation with a SHA-1 core is also significantly more secure than the LSIB-based scheme with 128-bit keys.

VI. CONCLUSION

Effective access management methods for on-chip instrumentation are crucial to achieve system safety and security. These techniques are also necessary to meet various requirements on instrument accessibility throughout the lifetime of a chip. In this paper, we propose a secure access management scheme for reconfigurable scan networks compliant with IEEE Std 1149.1-2013 (JTAG) and IEEE Std 1687-2014 (IJTAG). This technique is based on a secure challenge-response protocol and provides fine-grained control over the accessibility of individual protected instruments. It requires only a slight modification of the original scan infrastructure and incurs marginal performance overhead. If more than just a few instruments need protection, our approach scales very well, has favorable cost, and does not cause routing issues.

ACKNOWLEDGEMENTS

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

REFERENCES

[1] 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.

[2] M. Abramovici, “In-System Silicon Validation and Debug,” *IEEE Design & Test of Computers*, vol. 25, no. 3, pp. 216–223, 2008.

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

[4] R. Baranowski, “Reconfigurable Scan Networks: Formal Verification, Access Optimization, and Protection,” Ph.D. dissertation, University of Stuttgart, 2014. [Online]. Available: http://elib.uni-stuttgart.de/opus/volltexte/2014/8982

[5] “IEEE Standard for Test Access Port and Boundary-Scan Architecture 1149.1-2013,” IEEE Computer Society, 2013, Test Technology Technical Committee of the IEEE Computer Society, USA.

[6] J. Rearick, B. Eklow *et al.*, “IJTAG (Internal JTAG): A Step Toward a DFT Standard,” in *Proc. IEEE International Test Conference (ITC)*, 2005, paper 32.4.

[7] F. Ghani Zadegan, U. Ingelsson *et al.*, “Access Time Analysis for IEEE P1687,” *IEEE Trans. on Computers*, vol. 61, no. 10, pp. 1459–1472, October 2012.

[8] J. Da Rolt, A. Das *et al.*, “Test versus Security: Past and Present,” *to appear in: IEEE Trans. on Emerging Topics in Computing*, 2014.

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

[10] L. Greenemeier, “iPhone Hacks Annoy AT&T but Are Unlikely to Bruise Apple,” *Scientific American*, Aug. 30, 2007, online: http://www.scientificamerican.com/article/iphone-hacks-annoy-at (accessed Feb. 24, 2014).

[11] B. Yang, K. Wu, and R. Karri, “Scan Based Side Channel Attack on Dedicated Hardware Implementations of Data Encryption Standard,” in *Proc. IEEE International Test Conference (ITC)*, 2004, pp. 339–344.

[12] E. Ebrard, B. Allard *et al.*, “Review of Fuse and Antifuse Solutions for Advanced Standard CMOS Technologies,” *Microelectronics Journal*, vol. 40, no. 12, pp. 1755–1765, 2009.

[13] O. Kömmerling and M. G. Kuhn, “Design Principles for Tamper-Resistant Smartcard Processors,” in *Proc. USENIX Workshop on Smartcard Technology (WOST)*. USENIX Association, 1999, pp. 9–20.

[14] L. Sourgen, “Security Locks for Integrated Circuit,” May 1992, US Patent App. 5101121 A.

[15] D. Hely, M. L. Flottes *et al.*, “Scan Design and Secure Chip [Secure IC Testing],” in *Proc. IEEE On-Line Testing Symposium (IOLTS)*, 2004, pp. 219–224.

[16] J. Lee, M. Tehranipoor, and J. Plusquellic, “A Low-Cost Solution for Protecting IPs Against Scan-Based Side-Channel Attacks,” in *Proc. IEEE VLSI Test Symposium (VTS)*, 2006, pp. 94–99.

[17] J. Lee, M. Tehranipoor *et al.*, “Securing Designs against Scan-Based Side-Channel Attacks,” *IEEE Trans. on Dependable and Secure Computing*, vol. 4, no. 4, pp. 325–336, Oct.-Dec. 2007.

[18] K. Agarwal, “Secure Scan Design,” Jun. 2011, US Patent App. 7,966,535.

[19] G.-M. Chiu and J.-M. Li, “A Secure Test Wrapper Design Against Internal and Boundary Scan Attacks for Embedded Cores,” *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 1, pp. 126–134, Jan. 2012.

[20] J. Dworak, A. Crouch *et al.*, “Don’t Forget to Lock your SIB: Hiding Instruments using P1687,” in *Proc. IEEE International Test Conference (ITC)*, 2013, paper 6.2.

[21] R. Buskey and B. Frosik, “Protected JTAG,” in *Proc. IEEE International Conference on Parallel Processing Workshops (ICCPW)*, 2006, pp. 405–414.

[22] C. Clark, “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)*, 2010, pp. 19–24.

[23] K. Rosenfeld and R. Karri, “Attacks and Defenses for JTAG,” *IEEE Design & Test of Computers*, vol. 27, no. 1, pp. 36–47, 2010.

[24] K.-Y. Park, S.-G. Yoo *et al.*, "JTAG Security System Based on Credentials," *Journal of Electronic Testing (JETTA)*, vol. 26, pp. 549–557, 2010.

[25] K.-Y. Park, S.-G. Yoo, and J. Kim, "Debug Port Protection Mechanism for Secure Embedded Devices," *IEEE Journal of Semiconductor Technology and Science*, vol. 12, no. 2, pp. 240–253, 2012.

[26] A. Das, J. Da Rolt *et al.*, "Secure JTAG Implementation Using Schnorr Protocol," *Journal of Electronic Testing (JETTA)*, vol. 29, no. 2, pp. 193–209, 2013.

[27] L. Pierce and S. Tragoudas, "Enhanced Secure Architecture for Joint Action Test Group Systems," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 21, no. 7, pp. 1342–1345, 2013.

[28] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*. CRC Press, 1996.

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

[30] R. Baranowski, M. A. Kochte, and H.-J. Wunderlich, "Securing Access to Reconfigurable Scan Networks," in *Proc. IEEE Asian Test Symposium (ATS)*, 2013, pp. 295–300.

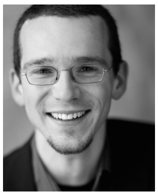
[31] R. Baranowski, M. A. Kochte, and H.-J. Wunderlich, "Access Port Protection for Reconfigurable Scan Networks," *Journal of Electronic Testing (JETTA)*, vol. 30, pp. 711–723, 2014.

[32] A. Crouch and J. Potter, "Protection of Proprietary Embedded Instruments," April 2011, US Patent App. 12/898,533.

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

[34] 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.

Rafal Baranowski received a M.Sc. degree in Electronics and Telecommunications from the Silesian University of Technology in 2007, and a Dr. rer. nat. (Ph.D.) from the University of Stuttgart in 2014. His research interests include hardware security and formal hardware verification.



Michael A. Kochte received a Diploma in Computer Science in 2007 and a Dr. rer. nat. (Ph.D.) from the University of Stuttgart in 2014. He joined the Institute for Computer Architecture and Computer Engineering of the University of Stuttgart in 2007. Apart from hardware infrastructure and security, his research interests comprise ATPG, fault and circuit simulation, and VLSI dependability.

Hans-Joachim Wunderlich is Director of the Institute of Computer Engineering (ITI) at the University of Stuttgart. He has authored and co-authored more than 200 publications in the area of test, reliability, and fault tolerance.