Protecting test port of embedded systems to keep data safe

Benjamin Dauphin
TIMA/AMfoRS
Grenoble, France

Abstract—Security is a growing concern for embedded system designers. Therefore a wide range of protection mechanisms have been developed in the recent years to counter malicious attacks. This work gives an overview and compares existing solutions concerning the protection of the test access port. Potential improvements are then proposed over some of the previous works.

Keywords—security, JTAG, IEEE 1687, DfT, test access port

I. INTRODUCTION

Malicious attacks are known on many examples of embedded systems aiming at, for example, stealing private data or encryption secret keys. In the past, protection measures where used only in sectors with security as a prerequisite (e.g., military applications) but during the last decade the trend towards security has gained mass market products as well. This need for security will continue, as systems containing private or potentially harmful information become ubiquitous (e.g., cars, home automation and Internet of Things).

In order to ensure the quality of products, testing is required. Hence, standards have been developed, like JTAG (IEEE 1149.1) or the more recent IEEE 1687 (Internal JTAG, adopted in 2014). But these testing interfaces give more observability and controllability, leading to the leaking of sensitive information. A solution would be to fuse the test port after testing, but this prevent diagnosis of faulty device. This also forbid any reconfiguration of the system, such as firmware update. Fusing is hence not a conceivable option for designers of set-top boxes and similar products.

This paper focuses on the protection of test and configuration access in embedded systems. A previous survey concerning test and security measures for PCK hardware implementations can be found at [1]. It should also be noted that threats are not limited to test access port. Examples of solutions regarding the security of communication between chips and protection against counterfeit IPs in MPSoCs designed to extract data from neighbor chips can be found at [2], [3] and [4].

The paper is organized as follows. Section 2 presents solutions to produce useless data for an attacker. Section 3 explores methods to completely lock the test port and Section 4 focuses on works related to the recent IEEE 1687 standard. Then Section 5 summarizes and compares the different solutions. Finally Section 6 proposes enhancements to some of the existing solutions.

II. MAKING TEST PORT DATA UNUSABLE FOR ATTACKERS

A common way to test an Integrated Circuit is the use of a scan chain. A scan chain is a shift-register of flip-flop in a circuit. Each flip-flop out is connected to the next flip-flop in, and the chain is connected to scan-in and scan-out pins. This gives great observability and controllability of the system as data can be inserted in or extracted from it. Over the years, several solutions have been developed to reduce or prevent the leakage of information using the scan chain.

One possibility is the scrambling of the scan chain. Scrambling consists in cutting a global chain in sub-chains and then changing the order of the sub-chains regularly and randomly if the test is deemed insecure. This can be implemented for example by requiring a key-register to be set at a specified value known only by trusted testers [5]. Scrambling does not prevent extraction of data but makes much more difficult for a hacker to unveil private data such as encryption keys because the extracted data is in inconsistent order. One of the drawbacks of this technique is an increase in complexity of routing with the addition of new connections between parts of the chain. Another one is added power consumption.

Another way is to add a control in the system managing the switch between normal execution and test mode. A reset is done when switching, preventing a hacker from reading execution data, and from inserting mischievous data via the scan chain. In [6] the authors propose a very simple controller masking the output in execution mode and preventing a user from executing too many cycles in test mode. It has virtually no impact on the design of the system. In [7] Hély et al. use a more extensive system, using the states' encoding in the Finite State Machine to check the validity and ensure the execution of reset at every mode switch. This approach is more secure as it cannot be bypassed, but introduces a consequent area overhead.

The authors of [8] propose a secure scan-based Design for Testability. They use the don’t-care bits in test vectors to store a verification key unique to each vector. If a vector is inserted in the system but does not have the correct key a random output is generated instead. The uniqueness of key and the fact that don't care bits are at different places depending on the vector makes this structure very secure. The testing procedure is no different from basic testing, excepting the presence of key in test vectors. This means that the fault coverage rate is not affected by this solution.

Finally statistical learning can also be used to detect illegitimate access to the JTAG port [9]. This technique reposes on the fact that an attacker behaves most of the time differently
than a legitimate user. For example, an attacker without prior knowledge of the chip will try to discover the undocumented functions added by the designers to test the chip. They will likely use illegal instructions or make sequence of instructions not usually done by an usual tester. The designer of the chip configures the decision tree by running different scenarios of usage of the instructions, marking which ones are accepted and which ones look like an attack. With the authors benchmark, this method achieves a very high detection accuracy (over 95%). This method requires further optimizations as current implementation increases the length of the critical path by about 20% in comparison to the original JTAG logic circuitry, making it not usable as-is online.

Solutions in [5]-[8] are based on producing less usable or no data when the tester is not deemed trustable, and solution [9] prevent access when the tester has a suspect behavior. This kind of protection is not always sufficient to really secure an Integrated Circuit. A recent work from J. Fan et al. demonstrated that test port security is important even in blank chip where no key could be retrieved [10]. They showed how a hacker could use the scan-chain not to directly get sensitive information, but as a tool to facilitate others ways of attack, such as Side-Channel Attacks.

### III. PREVENTING ATTACKERS TO GET ACCESS TO THE TEST PORT

In order to restrict access to the testing port only to trusted individuals several options are available. A first option is to use a Symmetric Key Cryptographic (SKC) algorithm to enable the test port, but this solution is not really usable in real-life: Either a single key is defined for all the systems and this key will be cracked or leaked, or each device gets its own key, creating the need to manage a database associating device number and key, which is not reasonable for mass-market products.

In [11], Park et al. propose a three-entity authentication protocol as a potential solution to this problem. The protocol is composed of two phases: credential generation and user authentication. In the first phase the tester requires from the JTAG system a challenge and device ID that they next submit to a server. The server then generates credentials associated with the device if the user is authenticated and authorized to test this device. Credential verification is done on the tested device using the response given by the server. After the credential issue phase, the tester can have access to the system using its credentials to calculate the response to a challenge generated by the device to protect the access of the test port. The generated credential also contain a privilege level to limit the access only to some parts of the system, depending on the user.

Amitabh Das et al. propose in [12] the use of public key cryptography as another solution to the problems raised in SKC-based security. Their implementation uses the ECC-based Schnorr protocol. A two-way authentication is made to ensure that both the tester and the board are legitimate. This work is compatible with the JTAG standard, and does not impact the test performance once the system is unlocked.

These two works are based on the use of an authentication server, which complexifies the testing process.

The authors of [17] developed an architecture with multiple privilege levels. The security mechanism is based on two modules. First the Secure Authentication Module (SAM) provides the unlocking protocol (for example a challenge-response protocol). Unlocking is required to obtain access to the JTAG instruction register. Then the Access Monitor allows or forbids an instruction depending on the user's rights. It controls the data shifted in by monitoring bits in the scan chain, for example bits in the opcode or in the memory address whose access is requested. If the user does not have sufficient privilege the update signal is set to 0, preventing the illegal operation. This method has little timing overhead after the unlocking procedure. Furthermore, the modules can be designed to allow reconfiguration of rights by user with the uppermost access level.

### IV. IEEE 1687 SEGMENT INSERTION BIT: PARTIAL LOCKING

The Segment Insertion Bit (SIB) is a module on the scan chain that can be opened to give access to other instruments, giving the possibility to design longer chains, as not all parts of it are present on the path at a given moment. Long chains are a problem because they require more cycles to extract or insert data so SIB makes testing simpler.

In [13] and [14], the authors propose Locking SIB (LSIB), a mechanism to lock the access of some parts of the scan path. LSIB is a SIB that opens using different bits (key bits) on the chain instead of just its own bit. LSIB can also be insensitive to other attempts of opening or closing if some bits (trap bits) on the chain have been set at a specific value at any given moment. These mechanisms can be completed with false LSIBs acting like a honey pot for an attacker who wants to increase the length of the chain to get access to sensitive information, or with switching LSIBs that give access to different parts of the test path with same length, making it impossible for an attacker to know whether the SIB is open or closed. It should be noted that the LSIB mechanism is “security by obscurity”, as it works on the assumption that the hacker does not know the details of the scan path but the tester does.

Baranowski et al. propose in [15] another SIB evolution: Secure SIB (or S²IB). Instead of using other scan chain bits, the S²IB is secured using a challenge-response protocol to authenticate the tester. When authenticated, the user gains access to a second pair shift and update register. Both the usual update register and the protected one have to be set at 1 to get access to the protected segment. In comparison to LSIB, S²IB is costly if no cryptographic primitive is present and reusable on the chip, but more efficient if so when a large number of instruments should be protected.

This paper completed their previous work in [16]. In this previous paper they proposed another way to prevent the access to a segment of a scan chain, using a sequence filter. This filter observes the sequence of operations (shift, capture, update) and the Test Data Input (TDI) port to allow or inhibit an operation requested by the user. If the access pattern is forbidden, the update signal is forced to 0, preventing the execution of the forbidden operation. This approach is similar to [9] (statistical learning) as both observe the user's behavior to determine if they should be trusted.

Using [15] and [16] in combination gives the possibility to develop a multi-level privilege test port with locking by S²IB and control of user access to the system with the sequence filter configuration depending on their privilege.
regarding user privilege than other locking-based mechanisms.

currently one of the most secure, but offers less flexibility
al. in [12] responds to this requirement. This solution is
detect counterfeit chips, the test-wrapper developed by Das et
controller from Da Rolt et al. [6] is the easiest and cheapest
is not a problem for testing and maintenance, the smart
attacks with a low surface overhead and low or no effect on the
constraint on the design. For situations where simple security is
wanted, scan-chain scrambling or Talatule et al. solution [8]
with a reasonable key-size are good ways to prevent generic

TABLE I. SUMMARY OF SECURITY MECHANISMS

<table>
<thead>
<tr>
<th>Autentication</th>
<th>Unlocking</th>
<th>Multi -level</th>
<th>Reconfigurable</th>
<th>Standard</th>
<th>Time overhead</th>
<th>Surface overhead</th>
<th>Impact on DFT flow</th>
<th>Diagnosable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scrambling [5]</td>
<td>user</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>Shifting key</td>
<td>none</td>
<td>Low</td>
<td>Yes</td>
</tr>
<tr>
<td>Smart controller [6]</td>
<td>none</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>Reset</td>
<td>none</td>
<td>Negligible</td>
<td>None Limited</td>
</tr>
<tr>
<td>FSM test controller [7]</td>
<td>none</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>JTAG</td>
<td>Reset</td>
<td>Medium</td>
<td>Limited</td>
</tr>
<tr>
<td>Key in vectors don’t-care bits [8]</td>
<td>user</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>none</td>
<td>none</td>
<td>Low’</td>
<td>Yes</td>
</tr>
<tr>
<td>Statistical learning [9]</td>
<td>none</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>JTAG</td>
<td>none</td>
<td>Medium</td>
<td>Yes</td>
</tr>
<tr>
<td>3-entities credentials [11]</td>
<td>user</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>JTAG</td>
<td>1* time credential generation, then authentication</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>PCK Schnorr [12]</td>
<td>user &amp; system</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>JTAG</td>
<td>Authentication</td>
<td>None</td>
<td>High</td>
</tr>
<tr>
<td>SAM and AM [17]</td>
<td>user</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>JTAG</td>
<td>Authentication</td>
<td>negligible</td>
<td>High</td>
</tr>
<tr>
<td>LSIB [13]-[14]</td>
<td>none</td>
<td>yes</td>
<td>possible^a</td>
<td>no</td>
<td>1687</td>
<td>Shifting key</td>
<td>None</td>
<td>Low’</td>
</tr>
<tr>
<td>SHB and sequence filter [15]-[16]</td>
<td>user</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>1687^a</td>
<td>Authentication</td>
<td>None</td>
<td>Depends^e</td>
</tr>
</tbody>
</table>

a. Depends on the key size
b. As LSIB is based on knowledge of the scan network structure, different knowledge could be given to different users.
c. The size of SIB is increased, but it is low compared to the size of the original chain
d. [16] is compatible with JTAG (IEEE 1149.1), IEEE 1500 and JTAG (IEEE 1687)
e. Low if there are reusable cryptographic primitives on the IC.

V. SUMMARY

These different approaches cover a wide range of security needs, and are summarized in Table 1. It summarize the different characteristics of security measures. The first five columns describe the type of measure and the others columns the cost of design, implementation and effects on testing.

Most of the security mechanism based on locking the access to the test port are costly to implement. Nonetheless, locking solution using the recent IEEE 1687 (Internal JTAG) standard have a smaller cost and are more flexible regarding the parts of the systems which should be secured.

Solutions without locking are less secure because of the vulnerability to Side Channel attacks. But these solution have a lower area overhead, making them suitable for systems where security is not a strong requirement or where surface is a strong constraint on the design. For situations where simple security is wanted, scan-chain scrambling or Talatule et al. solution [8] with a reasonable key-size are good ways to prevent generic attacks with a low surface overhead and low or no effect on the timing of the test. If the reduced controllability and observability due to the limitation of number of cycle executed is not a problem for testing and maintenance, the smart controller from Da Rolt et al. [6] is the easiest and cheapest solution to implement.

When authenticating the IC is important, for example to detect counterfeit chips, the test-wrapper developed by Das et al. in [12] responds to this requirement. This solution is currently one of the most secure, but offers less flexibility regarding user privilege than other locking-based mechanisms.

VI. PROPOSED ENHANCEMENTS

With this summary of strengths and weaknesses of studied solutions, it is possible to propose combination of solutions or potential improvements that could be made.

The Schnorr based solution [12] can be refined. Using this protocol with a sequence filter like in [16] would make a more flexible solution, offering multi-level access and reconfigurability. Also, using Schnorr as the unlocking protocol for the SAM in [17] would permit to authenticate the IC and not just user, thus creating a versatile and very secure solution to protect test port data.

Scrambling is a protection mechanism with a low cost, but is currently not compatible with the SIB mechanism (which changes the chain length). Scrambling might be made more compatible with IEEE 1687 by requiring opened SIB to have a length multiple of the sub-chain length, or by applying scrambling locally for segment hidden behind a SIB.

Finally, a small change to the 3-entities approach [11] can easily make it reconfigurable. It only requires to store user access policies in a rewritable memory accessible only for user with the highest level of access, similarly to Pierce and al. solution [17].

REFERENCES


S. D. Talatule, P. Zode, P. Zode. “A secure architecture for the design for testability structures” in VLSI Design and Test (VDAT), 2015, pp. 1-6


