UDK621.3:(53+54+621+66), ISSN0352-9045 Informacije MIDEM 37(2007)3, Ljubljana

ANALYSIS OF POTENTIAL ATTACK SCENARIOS FOR SYSTEMS WITH IEEE STD 1149.1 SECURITY EXTENSION

Anton Biasizzo

Jozef Stefan Institute, Ljubljana, Slovenia

Key words: test, IEEE Std 1149.1, security

Abstract: The paper addresses the security problems of boundary-scan design. Recently proposed security extension for IEEE Std. 1149.1 providing a locking mechanism is discussed. Possible attack scenarios are analysed. Complete attack time is calculated for different lengths of Key/Lock registers. For a large length of the Key/Lock registers it is practically impossible to perform a complete attack. Assuming that the attacker has some limited time interval to perform the attack, the probability of compromising the system is explored and the probability of successful attack within a given time interval is calculated. Test Access Port control logic with locking mechanism was implemented in Xilinx Spartan3 FPGA. The mechanism requires small hardware overhead and can be easily included in the IEEE Std 1149.1 test infrastructure.

Analiza možnih scenarijev vdora v sisteme z vgrajeno varnostno razširitvijo standarda IEEE 1149.1

Kjučne besede: preizkušanje, standard IEEE 1149.1, varnost

Izvleček: V zadnjem času postajajo aktualni problemi zagotavljanja varnosti v sistemih z vgrajeno preizkusno linijo (boundary-scan). Za vsako integrirano vezje z vgrajeno preizkusno linijo in vsak sistem zgrajen iz takšnih vezij namreč obstaja nevarnost vdora. Na panelni diskusiji konference ITC 2004, ki jo je moderiral E.J. Marinissen, so obravnavali možnosti vdora v sistem in kraje intelektualne lastnine preko preizkusne infrastrukture [1-5]. Predlagana je bila rešitev osnovana na kriptiranju preizkusnih podatkov, tako da se v integrirano vezje na vhodu preizkusne linije doda vezje za dekodiranje podatkov, na izhodu linije pa vezje za kodiranje podatkov. Pomanjkljivost takšne rešitve je v tem, da je že samo vezje za dekodiranje in kodiranje dokaj kompleksno sekvenčno vezje in je zanj potrebno zasnovati vgrajen samodejni preizkus. Kraja intelektualne lastnine pa ni edini problem povezan z varnostjo sistemov z vgrajeno preizkusno linijo. Standardi IEEE 1149.1, IEEE 1149.4 in IEEE 1500 predvidevajo izvajanje preizkusa sistema preko namenskega preizkusnega vodila. V nekaterih izvedbah je to preizkusno vodilo priključeno na računalnik, ki je daljinsko dostopen preko interneta. V takšnih primerih je omogočeno daljinsko preizkušanje in vzdrževanje sistema, hkrati pa obstaja možnost vdora v sistem s strani neavtoriziranih oseb. Ob vdoru v sistem lahko napadalec sproži izvajanje preizkusnega ukaza, ki zmoti normalno delovanje sistema in ima zato lahko katastrofalne posledice. Za preprečevanje dostopa neavtoriziranim osebam je bila predlagana [6] varnostna razširitev standarda IEEE 1149.1. V tem prispevku analiziramo možne scenarije vdora v sisteme z vgrajeno varnostno razširitvijo standarda. Obravnavana sta dva osnovna načina vdora. Za različne velikosti varnostnega mehanizma so izračunane verjetnosti vdora v danih časovnih okvirih. Na kratko je predstavljena tudi praktična izvedba mehanizma v vezju FPGA.

1. Introduction

In recent years, discussion about the security problems of systems incorporating scan design has emerged. Any chip that uses scan design and any system built around it provides access to the system's internal logic and may be vulnerable to hackers. Possible theft of intellectual property via scan test infrastructure was discussed at a panel discussion at ITC 2004 moderated by E.J. Marinissen /1/. R. Kapur proposed a solution based on data encryption to protect the data in scan chains /2/, /3/. The application of cryptographic algorithms in scan design chain is, however, not trivial. The logic implementing a cryptographic algorithm is itself a complex sequential circuit which requires some design-for-test (DFT) solution: a conventional way to solve the problem is by organizing the flip-flops in a scan chain. On the same conference B. Yang, K. Wu, and R. Karri presented a paper in which they demonstrated the vulnerability of the implementation of DES algorithm with inserted scan chain using Synopsys Test Compiler /4/. Some further research in this topics has been reported recently by D.Hely et al. /5/.

Beside possible theft of intellectual property, scan design can be potentially misused for breaking in the system which may lead to a serious damage. Scan design is often combined with the test infrastructure of DFT standards IEEE Std. 1149.1, IEEE Std. 1149.4 and IEEE Std. 1500. In some implementations of remote system maintenance, test access port (TAP) is interfaced to a computer connected to internet. An attacker may crack the computer system and get access to the test port. Executing EXTEST or some other pin-permission instruction during normal system operation could have catastrophic consequences in safety critical applications. In order to prevent unauthorised users to access the system via IEEE Std. 1149.1 TAP, a security extension for IEEE Std 1149.1 has been proposed /6/. It provides a locking mechanism that can be activated manually or automatically after a predefined time-out. The security extension requires small hardware overhead and allows full conformance with IEEE Std. 1149.1. The proposed solution is also applicable to IEEE Std. 1149.4. Similar to the approach reported in /7/, it can be included as an extension in full conformance with IEEE Std. 1149.4. In this paper we analyze potential attack scenarios for the

systems with implemented IEEE Std 1149.1 security extension. The results may be of interest to the designers and managers when making decisions about the level of security of the prospective products.

2. IEEE Std 1149.1 security extension

A chip with implemented IEEE Std 1149.1 infrastructure and security extension is shown in Figure 1. The locking mechanism is shown in more details in Figure 2. The security extension of IEEE Std 1149.1 includes two additional instructions: LOCK and UNLOCK. When LOCK instruction is active the TAP control logic maps all instructions (except UNLOCK) to a harmless BYPASS instruction until the UNLOCK instruction with valid key code is applied.

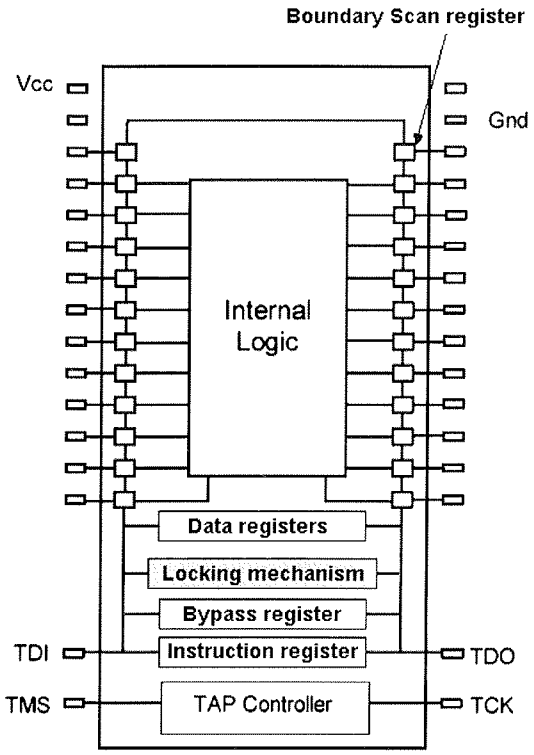

Fig. 1: IEEE Std 1149.1 infrastructure and security extension

The process of locking the TAP controller consists of the following steps:

1. LOCK instruction is entered into Instruction Register via TDI and decoded.
2. Lock Register and Key/Lock Shift Register are enabled.
3. The contents of the Key/Lock Shift Register is cleared at active Capture DR.
4. Lock code is entered into the Key/Lock Shift Register via TDI.
5. Lock code is transferred from the Key/Lock Shift Register to the Lock Register and Key Register is cleared at active Update DR.

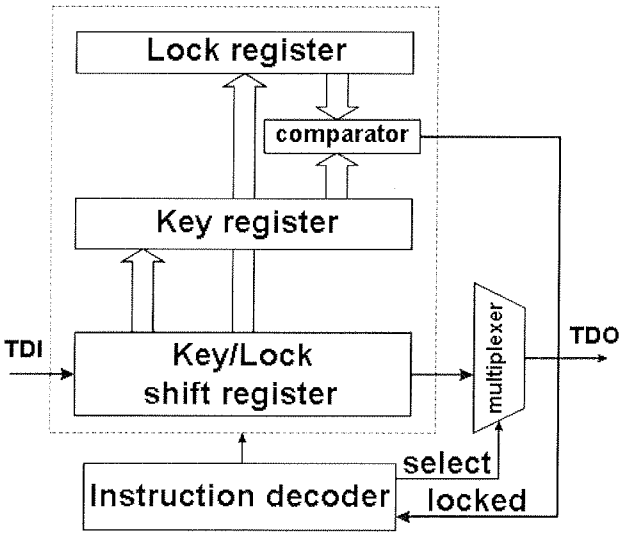

Fig. 2: IEEE Std 1149.1 locking mechanism

Comparator compares the contents of Lock Register and Key Register. If the contents are different, the Locked signal fed to the Instruction Decoder is activated. Consequently, the instruction decode logic maps all instructions except UNLOCK to the BYPASS instruction. This mapping is active until the Locked signal is released by executing UNLOCK instruction with the current key code.

Notice that the contents of Lock Register and Key Register are the same if the lock code is equal to zero. In this case, the TAP control logic remains unlocked.

The process of unlocking the TAP controller consists of the following steps:

1. UNLOCK instruction is entered into Instruction Register via TDI and decoded.
2. Key Register and Key/Lock Shift Register are enabled.
3. The contents of the Key/Lock Shift Register is cleared at active Capture DR.
4. Key code is entered into the Key/Lock Shift Register via TDI.
5. Key code is transferred from the Key/Lock Shift Register to the Key Register at active Update DR.

Comparator compares the contents of Lock Register and Key Register. If the contents are equal, it deactivates Locked signal and the next instruction entered via TDI can be executed. If the contents are different (i.e., wrong key code) the Lock signal remains activated and the TAP control logic remains locked.

3. Possible attack scenarios

We assume that the attacker has access to the Boundary-Scan infrastructure and is familiar with the locking mechanism described in the paper. However the attacker neither

knows the length of the Key/Lock registers nor the key code.

We can distinguish between two kinds of attacks to the locking mechanism:

1. Invasive attack, where the attacker's intention is to interfere with the operation of the circuit regardless if he/she leaves any traces of the attack (i.e., after the attack it is possible to deduce that someone interfered with the device).
2. Non-invasive attack, where the attacker first determines the lock code. The knowledge of the lock code enables the attacker to cover the tracks of the attack by locking the device with original lock at the end of the attack.

3.1. Invasive attack

3.1.1. Invasive attack scenario

We assume that the boundary-scan logic is in the Run-Test-Idle state. The attacker can easily change from any state to this state by maximum 6 additional clocks: 5 clocks to change to Test-Logic-Reset state (TMS=1) and one clock to change to Run-Test-Idle state (TMS=1) and this additional transitions are negligible part in comparison with the whole the attack.

For the invasive attack the attacker does not care about the value of the lock code. He/she merely tries to overwrite it by executing the LOCK instruction after each UNLOCK attempt.

The attack consists of:

1. Determining the length of the Key/Lock registers.

   The fact that the Key/Lock Shift Register is cleared at active Capture-DR can be also used for determining the length of the Key/Lock registers. The attacker executes the UNLOCK instruction and feeds values 1 to the input of the boundary-scan chain (TDI). By counting zeros at the output, the length of Key/Lock registers can be determined.
2. Repeating the following steps:
   - UNLOCK instruction, where the attacker generates a guess value of the lock code and tries to unlock the lock mechanism.
   - LOCK instruction, where the attacker overwrites the value of the Lock Register with 0. (Since the Key/Lock Shift Register is cleared at active Capture-DR the value 0 is the best choice for overwriting the Lock Register. In this way no Shift-DR cycles are required.) If previous attempt was unsuccessful the circuit remains locked and the LOCK instruction is ignored.

   It is worth noting that the attacker does not care in which step the Lock Register was successfully overwritten. The only goal is to have the value 0 placed in the Lock Register after the successive number of UNLOCK/LOCK executions.
3. Unlocking the circuit with 0 as the key and performing the malicious actions.

3.1.2. Invasive attack time analysis

Let L be the length of the Instruction Register and M the length of the Key/Lock registers, respectively. Let us denote the frequency of the boundary-scan clock (TCK) by *f*.

Determination of the length of Key/Lock registers starts in Run-Test-Idle state and stops in Select-IR Scan state of the boundary-scan test logic. It consists of:

- transition to Capture-IR state (3 cycles - TMS="110"),
- loading UNLOCK instruction (L cycles - TMS="0..0"),
- transition to Capture-DR state (4 cycles - TMS="1110"),
- loading values "1" to TDI and monitoring output TDO (M+1 cycles - TMS="0..0"),
- transition to Select-IR Scan state (4 cycles - TMS="1111").

Complete determination of the length of Key/Lock registers consists of

L+M+12 (3+L+4+M+1+4) cycles of boundary-scan clock.

Unlock attempt starts in Select-IR Scan state and consists of:

- transition to Capture-IR state (1 cycle - TMS="0"),
- loading UNLOCK instruction (L cycles - TMS="0..0"),
- transition to Capture-DR state (4 cycles - TMS="1110"),
- loading unlock code (M cycles - TMS="0..0"), (first unlock code has one additional clock)
- transition to Select-IR Scan state (4 cycles - TMS="1111").

Complete unlock attempt consists of L+M+9 (1+L+4+M+4) cycles of boundary-scan clock.

Lock attempt also starts in Select-IR Scan state and consists of:

- transition to Capture-IR state (1 cycle - TMS="0"),
- loading LOCK instruction (L cycles - TMS="0..0"),
- transition to Capture-DR state (4 cycles - TMS="1110"),
- transition to Select-IR Scan state (4 cycles - TMS="1111").

Complete lock command consists of L+9 (1+L+4+4) cycles of boundary-scan clock.

In the final UNLOCK command the unlock code is 0 and since Key/Lock Shift Register is cleared at Capture-DR state there is no need to load unlock code. It consists of:

- transition to Capture-IR state (1 cycle - TMS="0"),
- loading UNLOCK instruction (L cycles - TMS="0..0"),
- transition to Capture-DR state (4 cycles - TMS="1110"),

- transition to Run-Test-Idle state (3 cycles - TMS="110").

The length of the final unlock code is therefore L+8 cycles of boundary-scan clock.

The length of the complete attack must cover all possible lock codes. Since the length of Key/Lock registers is M there are 2^M possible codes and complete length of the attack is

Length = 2^M(L+M+9+L+9)+L+M+12+L+8 = 2^M(2L+M+18)+2L+M+20

The total time of the attack is given by

$$t = \frac{2^M (M + 2L + 18) + 2L + M + 20}{f}$$

Let us assume that the length of the Instruction Register is 8 bit and that the boundary-scan clock frequency is 100MHz. Some results for the time required for the complete attack for different lengths of Key/Lock registers are given in the table below.

M	time
8	108 μs
16	33 ms
32	2835 s
48	7.3 years
56	2056 years
64	573000 years

3.1.3. Incomplete attack and the probability of the successful attack

For a large length of the Key/Lock registers it is practically impossible to perform a complete attack. Suppose that the attacker has some time interval to perform the attack. Let us estimate the probability that the system will be compromised.

First we determine the number of unlock codes that the attacker can exploit in a time interval of length t:

$$N = \frac{t \cdot f - 2L - M - 20}{M + 2L + 18}$$

The probability that the system will be compromised is

$$p = \frac{N}{2^M} = \frac{t \cdot f - 2L - M - 20}{2^M (M + 2L + 18)}$$

Let us, determine the probability that the system will be compromised if the time interval is one hour for previous example

M	probability
32	100 %
36	7.5 %
40	0.44 %
48	0.0016 %
56	$5.6\ 10^{-8}$
64	$2\ 10^{-10}$

From the above equations we can estimate the lower bound of the length of the Key/Lock registers that would assure the required system security for a given time interval.

$$M = \log_2\left(\frac{t \cdot f - 2L - M - 20}{p\,(2L + M + 18)}\right) =$$
$$= \log_2(t \cdot f - 2L - M - 20) -$$
$$- \log_2 p - \log_2(2L + M + 18)$$

The derived equation cannot be solved analytically. Yet, the effect of the 2L+M+21 cycles is negligible and can be omitted. Since we are determining the lower bound of the length of the Key/Lock registers M we can replace the term (M + 2 L + 20) by s smaller value (2 L + 20), which gives

$$M_{(est)} = \log_2(t \cdot f) - \log_2 p - \log_2(2L + 18)$$

Minimal length of the Key/Lock registers for several probabilities that the system will be compromised in one day are given in the table below.

probability	$M_{(est)}$
100 %	37.9
10 %	41.2
1%	44.5
0.1%	47.9
10^{-4}	51.2
10^{-5}	54.5
10^{-6}	57.8

3.2. Non-Invasive attack

3.2.1. Non- Invasive attack scenario

Assumptions for non-invasive attack are the same as in the case of invasive attack: the Boundary-Scan logic is in the Run-Test-Idle state. If this is not the case the attacker can easily change from any test logic state to Test-Logic-Reset in 5 test clocks and to Run-Test-Idle in one additional clock.

The goal of the attacker is not just to get access to the Boundary-Scan test infrastructure but also to cover his/her tracks. This can be achieved only by determining the lock code. In order to determine the lock code the attacker has to check if the boundary-scan test logic is unlocked after every unlock command. This can be accomplished by determining the length of the data path for any restricted instruction, which uses data register longer than bypass register. When the boundary-scan test logic is locked bypass register is placed in the data path instead of the protected data register. The LOCK instruction is an obvious choice since it can be used also to erase the lock code. As was the case with the evaluation of the lock register length the attacker can take advantage of the fact that the value of the lock shift register is cleared prior the shifting.

The Non-Invasive attack consists of:

1. Evaluation of the key/lock register length as described in the case of Invasive attack.

2. Repeating of the following steps:
   - UNLOCK instruction with the guessed value of the lock code,
   - LOCK instruction and check if the length of the data path:
     - if the length of the data path is 1 then the boundary-scan test logic remains locked and step 2 is repeated with new guess value,
     - if the length of the data path is longer than 1 then the guess value is correct lock code. Use 0 as the new lock code (unlock the boundary-scan test logic) and stop the attack.

After the circuit exploitation with unlocked boundary-scan test logic the test logic can be locked with the original lock code in order to cover the track of the intrusion.

3.2.2. Non-Invasive attack time analysis

Let L be the length of the Instruction Register and M the length of the Key/Lock registers, respectively. Let us denote the frequency of the boundary-scan clock (TCK) by *f*.

Non-Invasive attack is performed in the following steps:

1. Evaluation of the length of the key/lock register starts in Run-Test-Idle state and stops in Select-IR Scan state of the boundary-scan test logic. It consists of:
   - transition to Capture-IR state (3 cycles - TMS="110"),
   - loading UNLOCK instruction (L cycles - TMS="0...0"),
   - transition to Capture-DR state (4 cycles - TMS="1110"),
   - loading values "1" to TDI and monitoring output TDO (M+1 cycles - TMS="0...0"),
   - transition to Select-IR Scan state (4 cycles - TMS="1111").

   This step is done in L+M+12 test cycles.
2. Unlock attempt starts at Select-IR Scan state of the boundary-scan test logic and consists of:
   - transition to Capture-IR state (1 cycle - TMS="0"),
   - loading UNLOCK instruction (L cycles - TMS="0...0"),
   - transition to Capture-DR state (4 cycles - TMS="1110"),
   - loading unlock code (M cycles - TMS="0...0"),
   - transition to Select-IR Scan state (4 cycles - TMS="1111").

   This step is done in L+M+9 test cycles.
3. It is followed by data path length check, which consists of:
   - transition to Capture-IR state (1 cycle - TMS="0"),
   - loading LOCK instruction (L cycles - TMS="0...0"),
   - transition to Capture-DR state (4 cycles - TMS="1110"),
   - length check (1 cycle - TMS="0", TDI="1"):
     a. if the length of data path is 1 ("1" shifted out of TDO)
        - make transition to Select-IR Scan state (4 cycles - TMS="1111"),
        - repeat step 2 with new guessed unlock code,
     b. if the length of data path is greater then 1 ("0" shifted out of TDO):
        - load lock code 0 (M cycles - TMS="0...0", TDI="0...0"),
        - transition to Run-Test-Idle state (3 cycles - TMS="110").

This step is done in L+10 test cycles, when the guessed lock value is wrong and in L+M+9, when the guessed lock value is correct.

After successful attack the Boundary-Scan test logic is in Run-Test-Idle state and current lock code is 0 (i.e. test logic is unlocked).

In the worst case for the attacker all 2^M codes are tried where 2^{M-1} were wrong. The length of such attack is:

$$\begin{aligned} Length &= 2^{M-1}\cdot(L+M+9+L+10)+ \\ &+(L+M+9+L+M+9)+2 = \\ &= 2^M\cdot(2L+M+19)+M+1 \end{aligned}$$

The total time of the Non-Invasive attack is given by

$$t=\frac{2^M(M+2L+19)+M+1}{f}$$

With the same assumptions of the length of the Instruction register and boundary-scan clock frequency as in the case of the Invasive attack the required time for the complete attack with respect to different lengths of Key/Lock registers are given in the table below.

M	time
8	110 μs
16	33.4 ms
32	2880 s
48	7.4 years
56	2080 years
64	580000 years

3.2.3. Incomplete Non-Invasive attack and the probability of successful attack

As was the case for Invasive attack it is practically impossible to perform complete Non-Invasive attack of boundary-scan test logic with large Key/Lock registers. It neither makes sense, since the attack is stopped as soon as the valid key is found. Therefore more useful measure of the security strength of the circuit is given by the probability that the system will be compromised in the given time span.

First we determine the number of unlock codes that the attacker can exploit in given time interval *t* :

$$N = \frac{t \cdot f - M - 1}{M + 2L + 19}$$

The probability that the system will be compromised is

$$p = \frac{N}{2^M} = \frac{t \cdot f - M - 1}{2^M (M + 2L + 19)}$$

The probabilities that the system will be compromised in the time interval of one hour for the previous example is given in the following table

M	probability
32	100 %
36	7.4 %
40	0.44 %
48	0.0015 %
56	$5.5\ 10^{-8}$
64	$2\ 10^{-10}$

From the above equations we can estimate the lower bound of the length of the Key/Lock register that would assure the required system security for given time interval:

$$M = \log_2\left(\frac{t \cdot f - M - 1}{p\,(2L + M + 19)}\right) =$$

$$= \log_2(t \cdot f - M - 1) - \log_2 p - \log_2(2L + M + 19)$$

This equation cannot be solved analytically yet the impact of M+1 is negligible and can be omitted. The term can be replaced with smaller value since we are estimating lower bound of the Key/Lock register length, which gives

$$M_{(est)} = \log_2(t \cdot f) - \log_2 p - \log_2(2L + 19)$$

In the following table the estimations of the lower bound of the Key/Lock register length are given.

probability	$M_{(est)}$
100 %	37.8
10 %	41.2
1%	44.5
0.1%	47.8
10^{-4}	51.1
10^{-5}	54.5
10^{-6}	57.8

4. Implementation

Test Access Port control logic with locking mechanism was implemented in Xilinx Spartan3 FPGA. The mechanism requires small hardware overhead and does not slow down the conventional boundary-scan tests. The implementation details for configurations with three different lengths of Lock Register / Key Register of are summarised in the table below. For comparison, the configuration data of the test logic without security extension is also shown in the table. In all cases, the length of the Boundary Data Register is 2 bits and the length of the Instruction Register is 4 bits.

	without security extension	included Lock Register / Key Register		
		8 bit	16 bit	32 bit
number of slices	34	48	61	92
number of slice Flip Flops	45	69	93	141
number of 4 input LUTs	61	86	107	149

5. Conclusions

Security extension of IEEE Std 1149.1 based on a locking mechanism was investigated: typical attack scenarios were considered and analysed. Test Access Port control logic with the locking mechanism has been implemented in Xilinx Spartan3 FPGA. The mechanism requires small hardware overhead and can be easily included in the IEEE Std 1149.1 test infrastructure.

References

/1./ E. J. Marinissen, moderator, "Security vs. Test Quality: Can We Really Only Have One at a Time?" Proceedings of the International Test Conference, 2004. pp. 1411.

/2./ R. Kapur, "Security vs. Test Quality: Are they mutually exclusive?" Proceedings of the International Test Conference, 2004. pp. 1414.

/3./ B. Yang, K. Wu, R. Karri, "Scan based side channel attack on dedicated hardware implementations of data encryption standard", Proceedings of the International Test Conference, 2004, pp. 339-344.

/4./ EE Times On Line, Latest News, Scan design called portal for hackers http://www.us.design-reuse.com/news/news8974.html

/5./ D. Hely, F. Bancel, M-L. Flottes, B. Rouzeyre, " Securing Scan Control in Crypto Chips", Journal of Electronic Testing, Theory and Practice, Vol. 23, No. 5, 2007, pp. 457 - 464.

/6./ F. Novak, A. Biasizzo, "Security extension for IEEE Std 1149.1", Journal of Electronic Testing, Theory and Practice, Vol. 22, No. 3, 2006, pp. 301-303.

/7./ U. Kac, F. Novak, F. Azais, P. Nouet, M. Renovell, "Extending IEEE Std. 1149.4 analog boundary modules to enhance mixed-signal test", IEEE Design & Test of Computers, Vol. 20, No. 2, 2003, pp. 32-39.

Anton Biasizzo
Jozef Stefan Institute, Jamova 39, 1000 Ljubljana
anton.biasizzo@ijs.si

Prispelo (Arrived): 28.03.2007 *Sprejeto (Accepted): 15.09.2007*