# Experimentations on scan chain encryption with PRESENT

Mathieu da Silva, Marie-Lise Flottes, Giorgio Di Natale, Bruno Rouzeyre

# Experimentations on Scan Chain Encryption with PRESENT

Mathieu Da Silva, Marie-lise Flottes, Giorgio Di Natale, Bruno Rouzeyre
LIRMM (Université Montpellier/CNRS)
161 rue Ada, Montpellier, France
{mathieu.da-silva,flottes,dinatale,rouzeyre}@lirmm.fr

***Abstract*—Crypto-processors are vulnerable to scan attacks. Using the scan chain, an attacker is indeed able to observe intermediate encryption states and steal secret data closely-related to the key. However, scan design is the most powerful mean for test and diagnostic purpose. Several countermeasure approaches have thus been proposed for securing scan designs while preserving test efficiency, diagnosis and debugging abilities. One solution is to encrypt test patterns thanks to extra block ciphers preventing control and observation of plain texts in the scan chain. The goal of this paper is to experiment this scan chain encryption approach on different designs in order to evaluate test efficiency and costs in terms of area and test time.**

***Keywords—Test and Security; Scan Attacks Countermeasure; Light Encryption***

## I. INTRODUCTION

Manufacturing testing allows distinguishing between fault-free and faulty circuits prior to shipping and ensures a high level of quality. Design-for-Testability (DfT) is a design approach aimed at improving circuit testability. The most popular DfT solution dedicated to logic circuits is the scan design. Scan design provides full control/observation of internal states at test time and thus reduces test pattern generation complexity, provides high fault coverage, simplifies fault diagnostic and support debug facilities. Scan design consists in replacing original registers by scan registers, i.e shift registers controlled/observed from/through primary IOs. However, an attacker can use observability and controllability offered by the scan registers to leak secret on the circuit, especially on crypto-processors [1][2]. Therefore, full control and observation have to be avoided to preserve data confidentiality.

Several countermeasures have been proposed to prevent these scan attacks [3][4][5][6][7][8][9][10]. The most common industrial practice consists in disconnecting test accesses after manufacturing test by using fuses. This low-cost solution does not impact manufacturing test activities but raises significant maintenance issues in the field. Indeed, even if probing techniques can be used for controlling and observing disconnected test signals, they also offer new perspectives for attackers.

A countermeasure that would prevent the use of fuses is proposed in [3]. It consists in resetting scan registers when the circuit switches from mission mode to test mode. Secret data stored on the round register of a crypto-processor for instance becomes unusable for an attacker. However, some scan attacks [11] rely only on the test mode, thereby sidestepping the reset countermeasure.

Further countermeasures rely on non-classical design flows. Technique presented in [4] consists in setting the scan flip-flops (FFs) order only in test mode. In mission mode, the scan FF order is dynamically and randomly set, preventing possible analysis of data scanned out. Test time and test generation effort are not affected by scan chain re-ordering, unfortunately, area overhead and power consumption increase in mission mode (+7%) is an issue. Another technique to limit attack through scan chain observation is the use of a secure embedded comparator as proposed in [5]. Instead of the scan chain content, the circuit only outputs the comparison result between the test responses stored in the scan chain and the expected ones. Test time and test coverage are not impacted but diagnostic complexity increases. Therefore, debug with secure comparator cannot be done easily. Another solution is the use of a different DfT approach, the Built-In Self-Test (BIST) [6]. It limits external scan chain control and observation but compromises diagnostic since the test procedure returns a compressed procedure.

Others solutions consist in using secure test wrappers [7]. Only trusted user can access to test facilities thanks to a locking mechanism that stops the attacker if she/he cannot provide the test key. Secure protocols require managing test keys. To protect the test wrapper, the authors in [8] propose to encrypt the content of the JTAG communication using stream cipher. To guarantee a high security, a hash function and a message authentication code are needed, affecting area cost.

The countermeasure described in [9] consists in using light block ciphers for scan encryption. Assuming a crypto-processor embedded with key management and storing, the same key storage present in the circuit under test is used to store the scan chain encryption key. No additional key management policy is needed, compared to a solution such as [10] where the solution implies to manage the key that is shifted in the scan chain for tester authentication. With scan chain encryption, trusted users have full control and observation for test and debug facilities, while an attacker cannot control the data scanned in the scan chain, nor observe circuit internal states because of the encryption. This solution does not cause any constraint on debugging, it is ensured without authentication protocol compared to countermeasures based on password or challenge/response, which involves shifting operations before any debugging operation.

This paper intends to evaluate costs involved by the solution proposed in [9] on several circuits examples. Two variants of the original scheme are evaluated showing that test time overhead due to scan chain encryption can be largely offset with low cost extra DfT.

The remainder of this paper is organized as follow. Section II presents the scan chain encryption approach. Section III details the cost of the proposed secure test scheme on several examples. Eventually, Section IV concludes the paper.

## II. PRESENTATION OF SCAN CHAIN ENCRYPTION

The countermeasure proposed in [9] targets circuits embedding at least one crypto-core. We assume a tamper-resistant memory to store secret-keys and a key management policy (no additional key management policy is required).

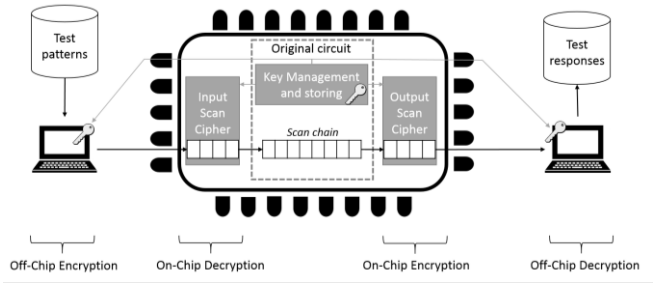


Fig. 1. Scan chain encryption scheme

The countermeasure implementation consists in adding light block ciphers at the input and the output of the scan chain (Fig. 1). Without the knowledge of the key, it is not possible to set the circuit to a desired state, nor to read plain circuit states. Input Scan Cipher prevents control-based scan attacks, Output Scan Cipher prevents observation-based scan attacks. This solution ensures a protection while keeping debug facilities. The developer wanting to debug an application can read and write on the registers of the processor.

The whole test procedure is described in Fig. 1. The first step consists in generating test patterns for the original circuit under test, and collecting expected test responses to these patterns from simulation. The generated test patterns are then encrypted off-chip. At test time, each encrypted test patterns is first scanned in the device then decrypted using the Input Scan Cipher before to be shifted in the scan chain of the Circuit Under Test (CUT). The test response to this pattern is stored in the scan chain of the CUT. While shifting out that test response, the Output Scan Cipher encrypts the data before circuit scan out. Eventually, encrypted test responses are decrypted off-chip to be compared with expected ones.

Each block cipher relies on an N-bits round register with two operating modes, parallel-load and shift. The parallel-load mode allows encrypting/decrypting CUT test data. The number of iteration for encryption/decryption depends on the implemented block cipher. The shift mode allows to serially deliver test data to the CUT scan chain (input block cipher), or to read out CUT scan chain content (output). N must be chosen according to the chosen block cipher.

Concerning test coverage, the original circuit is tested without any loss by using scan chain encryption since original test patterns are actually applied to the CUT after decryption. However, the additional test infrastructure has to be tested as well. Clearly, it is not possible to use classical scan design for the additional scan ciphers because, otherwise, scan attack would be possible on that ciphers.

## III. EXPERIMENTS

In the following experiments, two PRESENT block ciphers are used for the implementation of the Input/Output block ciphers, as also described in [12]. This decision was guided by the low implementation costs, for a sufficient obtained security level. If more security is required (for instance to cope with side-channel attacks) a solution is proposed in [13].

In this considered solution, the PRESENT encryption block size is 64 bits (N=64) and the encryption/decryption is done in 32 cycles. Experiments are conducted thanks to the synthesis tool Design compiler [14] and the ATPG tool TetraMAX [15].

### A. Area cost

Both scan ciphers share some common parts, especially the key expansion and the finite state machine controlling the operations. The total area of the proposed solution is 2081 combinational cells and 396 FFs. This area overhead is to be compared to the original circuit where the solution is implemented. Table 1 reports the overhead on 5 circuit examples: a triple-DES core, a pipelined AES core with the 128-bits and 256-bits version, a RSA 1024 bits core and a LEON3 processor. For each circuit, line Cell area reports the area of the original circuit after regular scan insertion (Scanned Circuit) and the overhead (%) induced by scan encryption. In case of very large circuit like LEON3 processor, the proposed solution induces only 0.57% overhead. This overhead represents 5.74% for the smallest circuit Triple-DES.

### B. Test coverage

In order to prevent any scan attack on the two extra scan ciphers, we prevent these extra blocs to be scanned out while storing an intermediate encryption state. The controller implemented to perform scan encryption on the original circuit involves full encryption (32 rounds) of every test pattern/test response for the original circuit, and control the scan enable signals allowing to shift PRESENT round registers. There is no "regular" scan chain procedure implemented for scan ciphers.

| Circuit | Triple-DES | | Pipelined AES128 | | Pipelined AES256 | | RSA 1024 | | LEON3 | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Scanned Circuit | Scan Encrypt. Overhead (%) | Scanned Circuit | | Scanned Circuit | | Scanned Circuit | | Scanned Circuit | |
| Cell Area DC estimation | 187 494 | +5.74 | 367 926 | +2.92 | 669 193 | +1.61 | 468 415 | +2.30 | 1 902 095 | +0.57 |
| Encrypt. Test Cov. | | 100% | | 100% | | 100% | | 100% | | 100% |
| #SFF | 8808=137×64+40 | | 7873=123×64+1 | | 12736=199×64 | | 16459=257×64+11 | | 107518=1679×64+62 | |
| #Patterns | 77 | | 246 | | 357 | | 2 393 | | 107 | |
| Test Cov. | 100% | | 100% | | 100% | | 100% | | 70% | |
| Test time (clock cycles) | 687101 | +0.31 | 1944877 | +0.81 | 4559845 | +0.01 | 39405239 | +0.33 | 11612051 | +0.004 |

Tab. 1. Cost of the proposed scan chain encryption for several circuits

The regular procedure would involve a control of their initial state and direct observation of their round register right after storage of an intermediate encryption state. Instead, the scan input-cipher's final state, i.e. the state achieved after one 32-rounds encryption, is only observable after several steps: (i) propagation through the scan chain of the original design, (ii) execution of one clock cycle for storage of the circuit response to that pattern, and (iii) full encryption through the output scan cipher. A similar procedure prevents the control of the output scan cipher: its initial state results from a decrypted pattern through the input scan cipher, and its final state is scanned out only after full encryption (32 rounds).

In order to provide test results on those extra ciphers we propose to evaluate test coverage achieved on those blocks thanks to the test procedure implemented for the original circuit. Test patterns, and respectively test responses, of the original circuit are indeed propagated and processed by the scan ciphers. These data are CUT dependent and thus can be regarded as random data w.r.t the scan ciphers' faults. Due to obfuscation and diffusion properties of cryptographic algorithms, it is expected that such circuit can be easily tested with random data [16] [17]. Both scan ciphers have thus been fault simulated while processing CUT test data. Input scan cipher is stimulated with encrypted CUT test data, the output scan cipher is stimulated with CUT test responses. Fault Coverage (FC) are reported in Table 1, line Encrypt. Test Coverage, and show that 100% FC have been achieved on the extra test circuitry on every case study. The number of test pattern/responses processed by the scan ciphers is reported in line #Patterns, and corresponds to the number of test patterns of the CUT. We can see that maximum fault coverage is also achieved with a short random sequence of only 77 patterns. Block ciphers are thus fully and "freely" tested during CUT test procedure.

### C. *Test time cost*

#### *1) Basic scan implementation*

Each scan cipher implements one N-bits round register to realize shifting and encrypting/decrypting operations. Every N clock cycles, an interruption of shift operations happens and the scan cipher must encrypt or decrypts the register content. In order to save test time, we implemented a second round register per scan cipher (see Fig. 2). While one of the two register is in shift mode in order to scan-in (respectively scan-out) CUT test data (resp. CUT test response), the other register is used for decryption/encryption.

With this optimization, Fig. 3.a shows the complete time diagram of shift operations with scan chain encryption. The circuit has F=*S*•*N*+*R* flip flops, where:

- F the total number of FF in the original circuit
- S the number of N-bit segments
- R = F modulo N

For the example presented Fig. 3, two N-bits segments (S=2) and a non-zero segment R compose the scan chain. Several patterns are used to test the circuit:

- $I_j^k$ (respectively $O_j^k$) the j[th] segment (with 0≤j≤S=2) of N bits of the k[th] test pattern (respectively response) provided to (and obtained from) the circuit
- E(x) the encrypted value of a segment x.

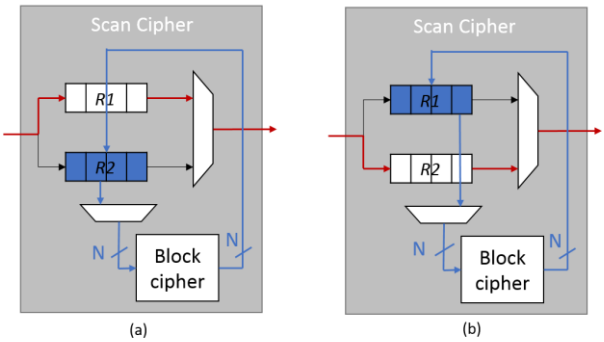


Fig. 2. Scan cipher optimized with two registers: (a) R1 receive shifted data while block cipher encrypts R2 content; (b) R2 receive shifted data while block cipher encrypts R1 content.

To ensure the test of the original circuit, it's necessary to pad patterns to have N-bits length segment, corresponding to block size cipher. That's why the first segment of each pattern $I_0^k$ is divided into a part of R bits $I_{0\,R}^k$ and a part of N-R bits $I_{0\,N-R}^k$. N-R bits complete each test pattern to have a length multiple of N. As it can be seen in the Fig. 3.a, at the step preceding the functional clock cycle (i.e., when the scan-enable signal is not anymore asserted and the flip-flops are going to sample the actual response of the circuit), the scan chain content is correctly filled with the input test pattern. However, concerning the first segment $I_0^1$, R bits are on the scan chain of the original circuit, while N-R are already entered in the Output Scan Cipher. At T=7×N+1, these N-R bits encrypted are shifted out and they have no impact on the response. The tester retrieves the same data sent in the input of the scan chain circuit.

Regarding the responses, two N-bits segments $O_0^k$, $O_1^k$ and a R-bits segment $O_{2\,R}^k$ form each response. The response is shifted out in the Output Scan Cipher for encryption. Fig 3.a presents the case of the first response $O^1$: the two first N-bits segments $O_0^1$ and $O_1^1$ are encrypted and shifted outside the circuit. For the remaining R bits response $O_{2\,R}^1$, the next input test pattern $I^2$ completes the register filling. Indeed the N-R bits $I_{0\,N-R}^2$ fill the output register with the R bits response $O_{2\,R}^1$ at T=8×N+1. The Output Scan Cipher encrypts the R-bits response with the additional part $I_{0\,N-R}^2$. The register content is then shifted out. On the received response, the tester can discard the N-R bits part. In the next operations, the others patterns are treated in the same way: the first N-R bits of input test pattern complete each last segment response of R-bits length.

Concerning the test time overhead, we have 2xN additional shift operation at the very beginning of the test procedure in order to feed the pipeline. The same additional shift operation is present at the end of the test to clean the pipeline. Moreover, we need N-R additional shift operations for each test pattern when R>0. More formally, by defining T the number of clock cycles for the original circuit to be tested without scan attack countermeasure, and K the overall number of test patterns, the number of clock cycles $T_f$ required to test the circuit with the encryption of the scan chain is shown in equation (1):

| **Pipelined AES-128 ($F = 7873 = 123 \times 64 + 1$)** | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Scan | Encrypt overhead (%) | Optimized version overhead (+63 FF) + Encrypt (%) | | | | | | | |
| #observation points per FF | | | 1 | 2 | 3 | 4 | 5 | ***6*** | 7 | 8 |
| #Patterns | 246 | 246 | 242 | 245 | 237 | 238 | 239 | ***235*** | 236 | 236 |
| Area | 367 926 | +2.92 | +3.10 | +3.17 | +3.23 | +3.30 | +3.36 | ***+3.43*** | +3.51 | +3.58 |

Tab. 2. Impact on number of observation points per flip-flops for test time optimization on Pipelined AES-128 circuit

$$T_f = \begin{cases} T + 2 \cdot 2N & if\ R = 0 \\ T + 2 \cdot 2N + (N - R)(K + 1) & if\ R > 0 \end{cases} \quad (1)$$

On each circuit, we determine the test time cost. Tab. 1 resumes the results: line #SFF reports the number F of scan FF, line #Patterns reports the number K of patterns needed to test the circuit, line Test Cov reports the test coverage of the original circuit. Line Test Time reports for each circuit the test time of the original implementation in terms of clock cycles, and the overhead induced by the encryption of both test patterns and test responses. In the case of pipelined AES 256 core, scan length 12736 FF is a multiple of N=64. Therefore, the test time overhead is only of 2x2N=256 cycles for scan-in initialization and last scan-out, which represents only 0.01% of the original test time. At the opposite, the pipelined AES 128 core has 7873=123×64+1 SFF. Therefore, the number of additional shift operations on each patterns is N-R=63 additional clock cycles. It's the worst case in terms of cost on each pattern. However, even in this case, the test time overhead is limited to 0.81%. For the others circuits, the number of additional clock cycles in each pattern is 24 for Triple-DES, 53 for RSA and 2 for LEON3 processor.

The number of patterns found by ATPG achieves only 70% of stuck-at fault coverage on that CUT because we stopped test pattern generation due to limitation in terms of memory allocation (line Test Cov in Table 1).

*2) Using extra bits for testability improvment*

Time diagram in Fig. 3.a, shows extra N-R bits on every test pattern for synchronization with the PRESENT cipher. These bits are scanned in the Output Scan Cipher and processed with following test responses without participating to the CUT test. We propose to use these extra bits for testability improvement. These N-R bits are thus stored into N-R dummy scan FFs appended to the original CUT scan chain instead of being scanned out in the output scan cipher. Fig.3.b details shift operations with N-R extra scan FFs. All 'last segment' of each pattern $O_2^k$ now includes N bits in this new implementation. It does not affect the test time compared to the former implementation (see III.C.1) since the number of shift operations remains the same.

The implementation involves extra costs in terms of area overhead. This cost is relative to the number of SFFs in the original CUT. For instance, LEON3 processor needs two extra FFs to pad its scan chain. The cost is only 2 FFs over 107518 FFs (<0.002%). In the worst case, 63 FFs are added to AES-128 core, i.e. 0.8% compared to the original scan chain with encryption. For Triple-DES (resp. RSA) circuit, dummy FFs represent an increase of 0.27% (resp. 0.32%) on the total scan chain. Tab 3. reports the overall cost of these designs compared to the original scan designs.

We now explain how these extra FFs can be used for testability improvement and, consequently, test time optimization.

Test point insertion is a classical Ad Hoc DfT procedure which consists in adding extra control or observation points to the circuit logic in order to improve its testability. Observation points in particular allows improving propagation of test responses to observable points, i.e. scanning FFs. Observability improvement usually results in reducing the test sequence length (less test patterns for same fault coverage).

The DfT tool Tetramax was used for selection of observation points in the circuit logic. We constrained the tool to use only the N-R extra FFs for testability improvement. After selection, observation points drives extra XOR trees ending on the proposed extra FFs, thus allowing their observation at test time. The XOR-trees configuration depends on the number of signals to observe. The number of test points per tree is user-defined and is ranging from 1 (only one observation point feeds the extra FF), to 8 (8 observation points drives an 8-input XOR tree feeding the extra FF).

To choose the best implementation, the eight cases are studied on each circuit. Tab. 2 presents the results for AES-128 core. As explained before, 63 extra SFF can be added to this circuit scan chain without affecting its test time. We iteratively experimented observability improvement with 1-to-8 observation points per XOR tree. For instance, with only one observation point per extra FF, the ATPG tool reduces the test sequence from 246 to 242 patterns. The best implementation corresponds to 6 observation points per extra FF, saving 11 patterns. Scan chain encryption applied on this implementation implies only 0.013% of test time overhead compared to 0.81% for scan chain encryption on original circuit. In other words, the extra test time due to the required synchronization with PRESENT encryption/decryption, and leading us to add extra shifts on every pattern, is compensated by the reduction of the total number of patterns to achieve the same fault coverage. The extra cost of scan chain encryption for test point insertion is +3.43% compared to original scanned design, with +2.92% for scan encryption without testability optimization, i.e without 63 extra FFs and 6-inputs XOR trees.

Experimental results on test time optimization and costs are presented in Tab. 3. For each circuit, the number of scanned flip-flops, the number of patterns, the test time (in clock cycles) and the area (cell area) are given for four versions: the original circuit with scan chain (row *Circuit*), the circuit with the scan chain encryption (row *Circuit+Encrypt*), optimized version with added scan FF connected to observability points (row *Optimized*) and optimized circuit with the scan chain encryption (row *Optimized+Encrypt*). Test time results for *Circuit+Encrypt* are compared to original

test time (*Circuit)* while results for *Optimized+Encrypt* are compared to test time with test point insertion (*Optimized*). Area of original scanned circuit (*Circuit*) is the comparison reference for both *Circuit+Encrypt* and *Optimized+Encrypt* versions. When two implementations lead to the same pattern optimization, we choose the implementation with the smaller impact in area.

Concerning optimized Triple-DES circuit, four observe points per FF are used on the 24 added SFF. Test time overhead decreases from 0.31% for scan encryption applied on original circuit by 0.038% for scan encryption applied on optimized circuit. However, the area cost of scan chain encryption increases from 5.74% when applied on original circuit to 5.87% when applied on optimized circuit. For LEON3 processor, 107 patterns achieve a test coverage of 70%. With 4 observe points on the 2 added SFF, this number of patterns decreases by 5 patterns. For scan encryption on LEON3 with test points, test time increases only by 0.002%, while for scan encryption on original LEON3, test time increases by 0.004%. The area cost is almost the same (0.57%) than the non-optimized scan chain encryption due to the large size of the CUT.

Unfortunately, insertion of test points doesn't allow reducing the number of test patterns on RSA. The AES-256 is not reported here since it is already optimized (its scan chain length is a multiple of 64). Extra test time for scan encryption is only 0.01% increase over non-encrypted scan test (see Tab.2).

| **Circuit** | | **#SFF** | **#Patt** | **Test time** | **Area** |
|---|---|---|---|---|---|
| Triple-DES | Circuit | 8808 | 77 | 687101 | 187 494 |
| | Circuit+Encrypt | 8808 | 77 | +0.31% | +5.74% |
| | Optimized | 8808+24 | 74 | 662730 | |
| | Optimized+Encrypt | 8808+24 | 74 | +0.038% | +5.87% |
| Pipelined AES 128 | Circuit | 7873 | 246 | 1944877 | 367 926 |
| | Circuit+Encrypt | 7873 | 246 | +0.81% | +2.92% |
| | Optimized | 7873+63 | 235 | 1873131 | |
| | Optimized+Encrypt | 7873+63 | 235 | +0.013% | +3.43% |
| RSA 1024 | Circuit | 16459 | 2393 | 39405239 | 468 415 |
| | Circuit+Encrypt | 16459 | 2393 | +0.33% | +2.30% |
| | Optimized | 16459+53 | 2393 | 39532121 | |
| | Optimized+Encrypt | 16459+53 | 2393 | +0.001% | +2.51% |
| LEON3* | Circuit | 107518 | 107 | 11612051 | 1902095 |
| | Circuit+Encrypt | 107518 | 107 | +0.004% | +0.57% |
| | Optimized | 107518+2 | 102 | 11074662 | |
| | Optimized+Encrypt | 107518+2 | 102 | +0.002% | +0.57% |

Tab. 3. Cost to use optimized scan chain encryption with test points regarding several circuits

*: for LEON3, test time and number of patterns are evaluated to obtain a test coverage of 70%

## IV. ACKNOKLEDGEMENT

This project has been funded by the French Government (BPI-OSEO) under grant FUI#20 TEEVA (Trusted Execution EVAluation).

## V. CONCLUSIONS

Scan chains offer facilities to realize scan attacks. A countermeasure proposed in [9] is based on the encryption of the scan chain content making test data unemployable for an attacker. Embedded scan cipher realize encryption and decryption operations with the key already present in the circuit. We need no additional key management policy. This secure solution can be used for test, diagnostic and debug.

In this paper, experimental results on several circuits show that this solution affects the original circuit with a marginal cost on area and test time. We have also proposed an optimization of the solution applicable on most of the circuits to minimize or even fully compensate extra test time due to longer shifts in the scan chain.

## REFERENCES

[1] B. Yang, K. Wu, R. Karri. Scan Based Side Channel Attack on Dedicated Hardware Implementations of Data Encryption Standard. In ITC, pp 339-344. IEEE, 2004.

[2] J. Da Rolt, G. Di Natale, M.-L. Flottes, B. Rouzeyre. Scan Attacks and Countermeasures in Presence of Scan Response Compactors. In ETS, pp 19-24. IEEE Computer Society, 2011.

[3] B. Yang, K. Wu, R. Karri. Secure Scan: A Design-for-Test Architecture for Crypto Chips. In IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 25, pp. 2287-2293, 2006.

[4] D. Hely, M.-L. Flottes, F. Bancel, B. Rouzeyere, N. Bernard, Scan design and secure chip. In Proc. IEEE Int. On-Line Test. Symp., pp. 219-224, 2004.

[5] J. Da Rolt, G. Di Natale, M.-L. Flottes, B. Rouzeyre. Thwarting Scan-Based Attacks on Secure-ICs With On-Chip Comparison. In Proc. IEEE Trans. on VLSI System, no. 22 pp. 947-951, 2013.

[6] M. Doulcier, M.-L. Flottes, B. Rouzeyre. AES-based BIST: Self-test, Test Pattern Generation and Signature Analysis. In 4th IEEE International Symposium on Electronic Design, Test & Applications, Hong-Kong, IEEE, pp.314-321, 2008

[7] Chiu G.-M.; Li J.; C.-M. A Secure Test Wrapper Design Against Internal and Boundary Scan Attacks for Embedded Cores, IEEE Trans. on VLSI System, vol 20, issue 1, p 126-134, 2010

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

[9] M. Da Silva, M.-L. Flottes, G. Di Natale, B. Rouzeyre, M. Restifo, P. Prinetto. Scan Chain Encryption for the Test, Diagnosis and Debug of Secure Circuits. 22nd ETS. 2017 [accepted for publication]

[10] J. Dworak, C. Crouch. Don't forget to lock your SIB: hiding instruments using P1687. International Test Conference (ITC), 2013.

[11] Sk Subidh Ali, Ozgur Sinanoglu, Samah Mohamed Saeed, and Ramesh Karri. New scan-based attack using only the test mode. In VLSI-SoC, pp 234-239. IEEE, 2013.

[12] A. Bogdanov, L.R. Knudsen, G. Leander, C. Paar, A. Poschmann, M.J.B. Robshaw, Y. Seurin, and C. Vikkelsoe, P. Paillier and I. Verbauwhede. PRESENT: An Ultra-Lightweight Block Cipher. CHES 2007, LNCS 4727, pp. 450–466, Springer-Verlag Berlin Heidelberg 2007

[13] A. Poschmann, A. Moradi, K. Khoo, C. Lim, H. Wang, S. Ling,. Side-Channel Resistant Crypto for less than 2,300 GE. J. Cryptology, vol. 24, pp 322-345, 2011

[14] Synopsys, Design Compiler. [https://www.synopsys.com/implementation-and-signoff/rtl-synthesis-test/dc-ultra.html]

[15] Synopsys, TetraMax. [https://www.synopsys.com/implementation-and-signoff/rtl-synthesis-test/test-automation/tetramax-atpg.html]

[16] G. Di Natale, M. Doulcier, M. L. Flottes, B. Rouzeyre. Self-Test Techniques for Crypto-Devices. IEEE Transaction on VLSI Systems, pp. 1-5, February 2010, Volume:18, Issue: 2.

[17] A. Schubert and W. Anheier. On random pattern testability of cryptographic VLSI cores. J. Electron. Test.: Theory Appl., vol. 16, no. 3, pp. 185–192, Jun. 2000.

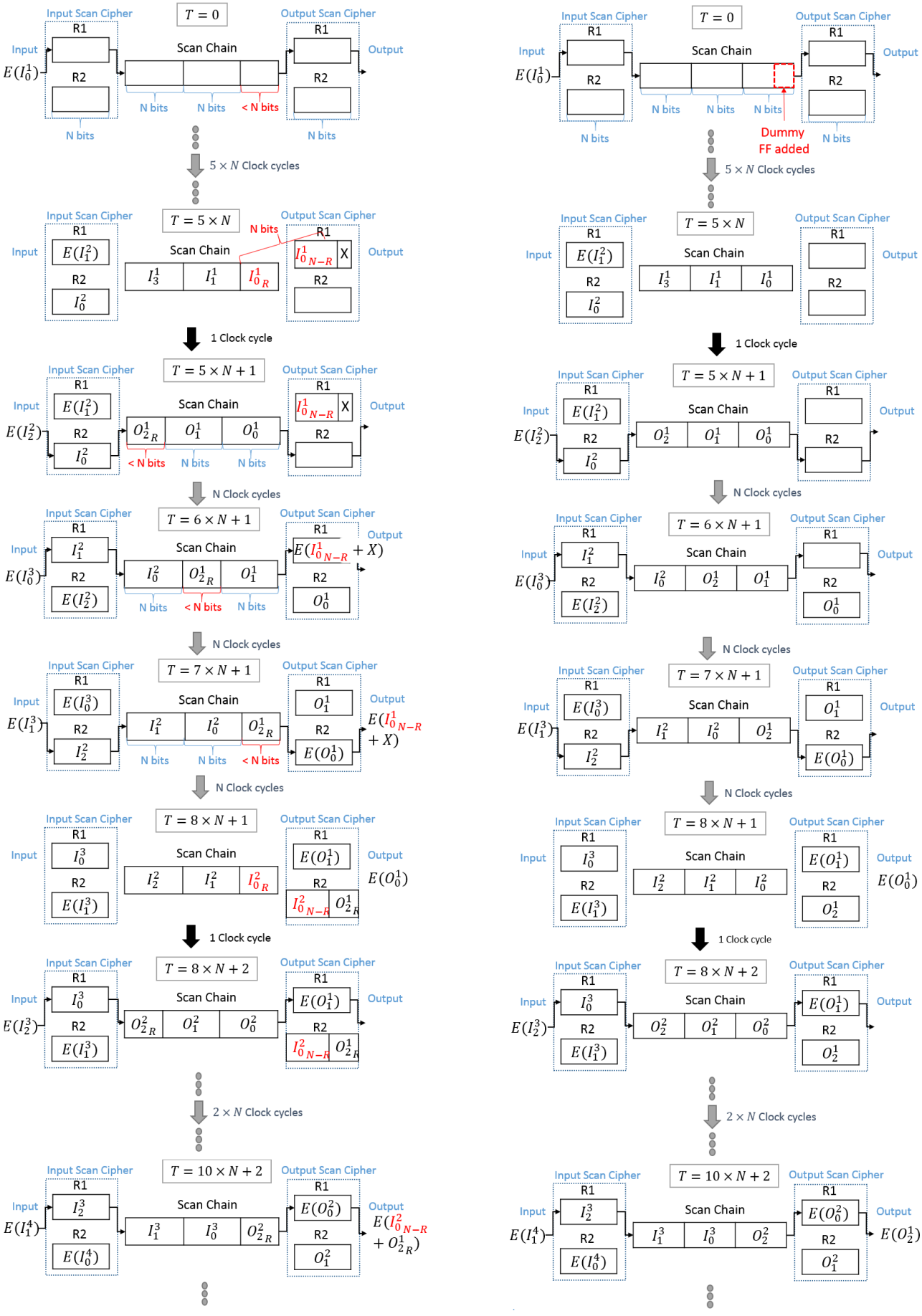


Fig. 3. Time Diagram of shift operations: (a) case where scan chain length isn't a multiple of N, (b) case where additional FF pad the scan chain