AES vs LFSR Based Test Pattern Generation: A Comparative Study

Marion Doulcier, Marie-Lise Flottes, Bruno Rouzeyre

AES vs LFSR based test pattern generation: A Comparative Study

M. Doulcier, M. L. Flottes, B. Rouzeyre

Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier
Université Montpellier II / CNRS UMR 5506
161 rue Ada, 34932 Montpellier CEDEX 5, France
{doulcier, flottes, rouzeyre}@lirmm.fr

***Abstract*—The massive integration of several functionalities leads to increased test times/test data volume. Additionally, test content for more advanced fault models increase the tester memory requirements. On the positive side, the presence of many cores in a system provides the opportunity of core testing each other. In this paper we evaluate the opportunity to use AES crypto-processors as test pattern generators. Several experiences are conducted on LFSRs and AES cores in order to compare their ability to generate pseudo-random test sequences.**

I. Introduction

Nowadays, more and more applications involve the exchange of confidential data (pay-tv, banking, phone applications, e-passport…). The test of such secure circuits is primordial to insure a high level of security since any malfunction may induce a security vulnerability. In order to answer to this confidentiality requirement, cryptographic operations are performed on exchanged data. Present secure circuits contain one or several cores dedicated to cryptographic operations. Cryptographic operations consist in ciphering plaintexts into cipher texts with the help of secret keys and in deciphering back the cipher texts into plaintexts when secret information has reached its destination.

Two approaches may be used to test these integrated circuits: an external scan-based test or a built-in self-test (BIST) approach. The external scan-based approach makes fully controllable and observable the memory cells included in the scan path and allows achieving very high fault coverage. However, this approach requires to store the deterministic test patterns in the Automatic Test Equipment, and moreover, the scan path can be used as a side channel to recover the secret key used during cryptographic operations. For instance in [1] and [2], authors demonstrate an attack to retrieve cryptographic key via scan chain information. This attack is realized in two steps; firstly the scan chain structure is determined and secondly the secret key is retrieved by shifting out circuit's internal states via the scan chain.

Conversely, the BIST approach prevents the observation of the data processed by the circuit. Generally, a pseudo-random sequence, generated for instance from a Linear Feedback Shift Register (LFSR) is applied to the modules under test, and the signature (compacted test response) obtained by mean of a Single or Multiple Input Signature Register (SISR or MISR) is compared to the expected one. One or several scan paths in the module under test may be fed from the LFSR but these scan paths are not visible from system's I/Os. However, BIST (or scan-BIST) approach requires the implementation of extra logic for test pattern generation and test response compaction.

This paper explores the possibility of substituting the classical LFSR-based test pattern generators by a cryptographic core already embedded in the circuit. In the context of secure circuits, this substitution allows to reduce the extra area due to test circuitry. The studied cryptographic core is an Advanced Encryption Standard (AES) core issued from the "Rijndael" algorithm and officially approved by the National Institute of Standards and Technology (NIST) in December 2001 [3].

The AES algorithm, its implementation and pseudo-random testability are presented in section II. Section III deals with the generation of pseudo-random test sequences from the AES and compares AES-based and LFSR-based test pattern generators in terms of test sequence randomness. Section IV presents several experiments on ISCAS'89 benchmark circuits. Section V discusses some issues. Finally conclusions are given in section VI.

II. AES: description and self-testability

The AES ciphers a 128 bits block plaintext into a 128 bits block cipher text with the help of a 128, 192 or 256 bits secret key.

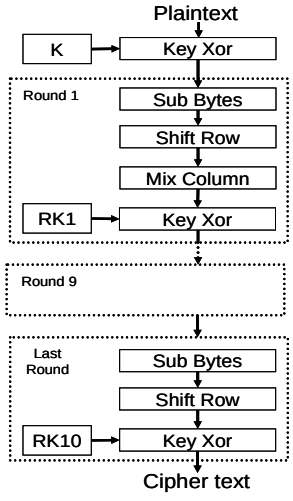

Figure 1: AES encryption principle

The 128 bits plaintext is considered as a 4*4 matrix of bytes. The algorithm consists in several rounds, each round being composed of operations (see figure 1): *Subbytes* is a substitution of text bytes using substitution tables called Sbox, *ShiftRows* consists in circular shifts on the matrix lines, *MixColumns* is a multiplication in the Galois field and *AddRoundKey* is a XOR operation between the partially ciphered text and the round key RK. RK is derived from the initial secret key K. According to the size of the initial key (128,192 or 256 bits), the number of rounds is respectively 10, 12 or 14. We will consider thereafter that the size of the key is 128 bits.

Before, to discuss the possibility of using the AES core as test pattern generator for some other cores in the secure system, it is important to note that this core can be self-tested. It is shown in [5] that random BIST is an efficient technique for testing cryptographic cores, i.e. high fault coverage can be achieved with short pseudo-random test sequences. This study is based on the statistical properties and qualities of the traditional operations (XOR, substitution, modulo….) involved in the cryptographic algorithms. These operations keep the randomness properties of the test data propagated through the circuit. In [6] a technique for self-testing the AES core is described. It consists in adding a feedback loop to the AES block. Data resulting from an encryption becomes the new data to cipher. Whatever the implementation of the AES blocks is, pipeline or iterative, the data path is completely tested. A 100% stuck-at fault coverage is achieved with 120 vectors i.e. after 120 encryption cycles. Moreover, when compared to standard pseudo-random BIST using extra LFSRs, this technique is more efficient since the area overhead as well as test time are reduced.

III. AES-BASED TEST PATTERN GENERATOR

A. TPG Implementation

Figure 2 represents the AES implementation for test pattern generation. After the tenth round, the ciphering result is stored in the register R and is conveyed to the extra-multiplexer feeding the first round. Same implementation is used when self-test is performed on the AES core.

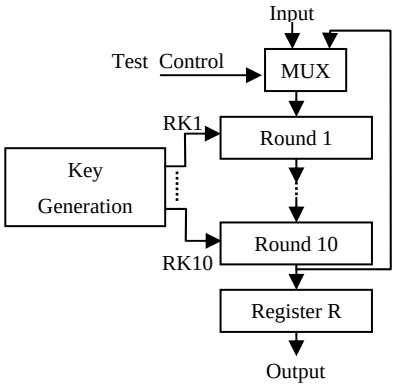

Fig. 2: AES implementation for Test pattern generation

The extra-multiplexer is controlled from the Test_Control signal and allows choosing between the single encryption of a plaintext (normal mode of operation) and iterative encryptions when AES is self-tested or used as test pattern generator (test mode). When in test mode, the first plaintext and the secret key are selected randomly. In short, after the first encryption is carried out, the cipher text represents the first generated 128 bits test pattern and becomes the new text to cipher. The process is iterated until a sufficient number of test patterns have been generated. The secret key is unchanged all along the generation process.

B. Evaluation of sequence randomness

Due to area constraints, BIST is generally synonymous of pseudo-random testing. Actually, implementing a test pattern generator delivering deterministic test sequences is generally too expensive except for testing regular structures (e.g. memories). Because we want to use the AES core as a pseudo-random test pattern generator, we study the randomness of sequences generated from this core when implemented as described in the preceding section.

The statistical test suite defined in NIST [7] allows evaluating the randomness properties of a data sequence.

This suite is composed of 15 statistical tests:

Monobit Test (Freq): The purpose of this test is to determine whether the number of ones and zeros in a sequence are approximately the same as would be expected for a truly random sequence.

Block Frequency Test (BlkFreq): The purpose of this test is to determine whether the number of ones and zeros in each of *M* non-overlapping blocks created from a sequence appear to have a random distribution.

Cumulative Sums Forward Test (CuSum): The purpose of this test is to determine whether the sum of the partial sequences occurring in the sequence under evaluation is too large or too small.

Runs Test (Runs): The purpose of this test is to determine whether the number of runs of ones and zeros of various lengths is as expected for a random sequence. In particular, this test determines whether the oscillation between such substrings is too fast or too slow.

Long Runs of Ones Test (LongRuns): The purpose of this test is to determine whether the longest run of ones within the sequence is consistent with the longest run of ones that would be expected in a random sequence.

Rank Test (Rank): The purpose of this test is to check for linear dependences among fixed length substrings of the original sequence.

Discrete Fourier Transform (Spectral) Test (DFFT): The purpose of this test is to detect periodic features (i.e., repetitive patterns that are near each other) in the sequence that would indicate a deviation from the assumption of randomness.

Aperiodic Templates Test (Aperiodic): The purpose of this test is to reject sequences that exhibit too many occurrences of a

given non-periodic (aperiodic) pattern.

Periodic Template Test (Periodic): The purpose of this test is to reject sequences that show deviations from the expected number of runs of ones of a given length.

Universal Statistical Test (Univ): The purpose of the test is to detect whether or not the sequence can be significantly compressed without loss of information. A compressible sequence is considered to be nonrandom.

Approximate Entropy Test (Apen): The purpose of the test is to compare the frequency of overlapping blocks of two consecutive/adjacent lengths (*m* and *m+1*) against the expected result for a normally distributed sequence.

Random Excursion Test (Random): The purpose of this test is to determine if the number of visits to a state within a random walk exceeds what one would expect for a random sequence.

Random Excursion Variant Test (Random): The purpose of this test is to detect deviations from the distribution of the number of visits of a random walk to a certain state.

Serial Test (serial1 & serial2): The purpose of this test is to determine whether the number of occurrences of *m*-bit overlapping patterns is approximately the same as would be expected for a random sequence.

Linear Complexity Test (LinComp): The purpose of this test is to determine whether or not the sequence is complex enough to be considered random.

If all these empirical tests are validated for a sequence, this one will be considered as a random sequence.

This statistical test suite has also been used to evaluate the secure property of cryptographic algorithms. In [8], it has shown that the AES algorithm returns random results at the end of the third round, thus the cipher text does not present any correlation with the plaintext.

The analysis performed in [9] shows that the AES is a good 128 bits words random generator for simulation purpose. For instance this property allows using the AES to realize a stream cipher. In the context of using AES as a test pattern generator, not only the randomness of the sequence of 128-bits words has to be studied, but also the randomness of the individual bitstreams as well as the randomness of sequence of n-bits words, with n<128 (section III.c). As a matter of fact, not all the outputs of the AES will be connected to the device under test.

C. Randomness properties: AES vs LFSR

In this section, we compare in term of randomness the sequences generated from the AES with those generated from type 1 (external Xor gates) and type 2 (internal Xor gates) LFSRs.

Type 1 and type 2 LFSRs have been implemented from 128 stages registers and a primitive polynomial $p(x) = x^{128}+x^{29}+x^{27}+x^{2}+1$. The number of stages has been chosen to be in accordance with the bitwidth of the AES output.

Three comparisons have been performed. First one compares randomness properties of bitstreams generated on a unique serial output of the test pattern generators (TPGs) (Figure 3.a). The second comparison deals with the randomness of sequences of 128 bits words generated on 128 bits TPG outputs (Figure 3.b). The last comparison deals with the randomness of sequences of 16 bits words generated on 16 randomly selected bits among 128 TPG outputs (Figure 3.c)

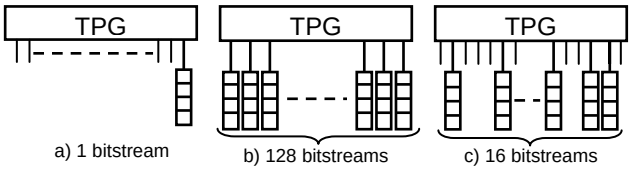

Fig. 3: Test data issued from AES

1. Bitstream randomness

We first compare the bitstream obtained on the less significant bit output of the AES core with the one obtained at the serial output of the type 1 and 2 LFSRs. The LFSRs seeds and the initial plaintext and the key for the AES have been randomly chosen. The length of the bitstream is arbitrarily fixed to $1.5*10^6$ bits.

The respective randomness of the bitstreams is compared thanks to the NIST statistical test suite.

Table 1: Statistical test results

	AES	LFSR type1	LFSR type2
1 : Freq	0.110981	0.000000	0.002560
2 : BlkFreq	0.267765	0.000000	0.441504
3 : CuSum	0.103183	0.000000	0.003386
4 : Runs	0.999049	0.000000	0.143622
5 : LongRuns	0.079787	0.000000	0.965931
6 : Rank	0.820208	0.000000	0.526598
7 : DFFT	0.642256	0.000000	0.810512
8 : Univ	0.845498	0.000000	0.244026
9 : Apen	0.189886	0.000000	0.637473
10 : Serial1	0.400669	0.000000	0.572199
11 : Serial2	0.227037	0.000000	0.855465
12 : LinComp	0.543506	0.000000	0.000000
13 : Aperiodic	0.453813	0.000000	0.499631
14 : Periodic	0.336229	0.000000	0.393849
15 : Random	0.517951	0.000000	0.000000

Table 1 reports the normalized results of every test for the 3 bitstreams. According to [7], a bitstream pass the test if the normalized result is greater than 0.01. In such case, the random criterion is confirmed with a level of significance of 99%.

It can be seen that the AES bitstream passes all the tests, i.e. the AES generates a "perfect" random sequence.

Conversely, the sequence obtained with help of the type 1 LFSR does not pass any test. One of the explanations is that the length of the bitstream is very short compared to the length of LFSR cycle ($1.5*10^6$ vs 2^{128} -1).

The sequence obtained with the type 2 LFSR (internal XORs) passes 11 tests over 15 (tests 1, 3, 12 and 15 fail).

On this example, it can be observed that a bitstream generated by the AES has better randomness properties than the ones obtained from a type 2 LFSR.

Those experiments have been repeated for several bitstreams obtained by changing the initial plaintexts and the crypto-keys, the LFSR seeds and/or by the length of the observed bitstreams. Similar results have been obtained for all experiments.

In spite of lack of theoretical proof, these experimental results indicate that AES-generated bitstreams have better randomness properties than LFSR-generated ones. Thus, looped AES module seems to be as good TPGs as LFSRs and could be used for testing sequential circuits equipped with a single scan chain.

2. N-bits words sequence randomness

In the previous section, we studied the randomness of bitstreams generated on only one output bit of the TPGs, i.e. the case for which the TPG feeds only one input of the circuit under test. However when the TPG feeds several (n) inputs of the circuit (like in the STUMP architecture [10]), the randomness of the sequence composed of n-bits words has to be questioned. Unfortunately, in our knowledge, no randomness criterion for multi-bits words has been adopted yet. Thus, we proceeded as follows:

1/ we questioned the randomness of every bitstream as before (there n bitstreams to study).

2/ we measured the (none)-correlation of n-bits words by:

- counting the number of occurrences of each word,
- looking at the distribution of bits set to 1 in the n-bits words,
- evaluating the randomness of the sequence made up of the successive n-bits words joined end to end. This randomness is questioned with help of the NIST suite.

We first performed the experiments with 128-bits words for the case where the TPG feeds 128 scan chains in the circuit under test. Secondly we consider circuits that contain fewer scan chains. For instance for 16 scan chains, 16 output bits are randomly selected among the 128 available bits on TPGs output. In the following experiments, bits 2, 18, 25, 38, 40, 41, 59, 71, 75, 80, 98, 100, 101, 110, 111 and 125 have been selected. As before, the length of the sequences is $1.5*10^6$.

Due to the poor results obtained with type 1 LFSR, we only compare in the following the sequences obtained from AES and type 2 LFSR.

a) Sequences of 128 bits words:

Each of the 128 bitstreams obtained on 128 output bits has been submitted to the NIST test suite as before. In order to analyse globally the results, we first compute the proportion of bitstreams that pass every test. It must be recalled that a bitstream pass a test if the normalized result is greater than 0.01 with a level of significance of 99%. In other words this means that a true random bitstream may lead to a result lower than 0.01 (and thus declared as non-random) with a probability lower than 1%. In order to alleviate this bias, we have computed an interval of confidence for which the set of bitstreams is considered as passing the randomness test. This interval can be computed as:

$$p \pm 3\sqrt{\frac{p \times (1-p)}{m}}$$

where m is the number of bitstreams and p is the confidence level (i.e. 99%). For m=128 and p=99%, the interval is [0.99-0.0263836, 1].

Figure 4 gives the proportion of bitstreams generated by the AES that passes the test. The two horizontal lines mark the limits of the interval of confidence. It can be seen that all bitstreams respect the 15 randomness criteria.

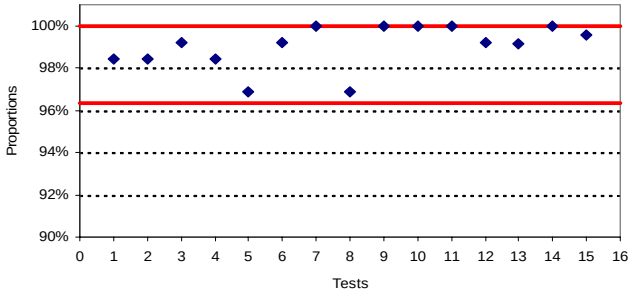

Fig. 4: percentage of AES bitstreams passing each test

Figure 5 reports the same information for the type 2 LFSR sequence. In this case, the sequence does not pass three of the tests.

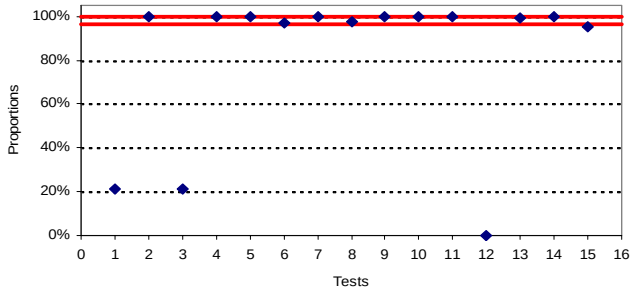

Fig. 5: percentage of type 2 LFSR bitstreams passing each test

Concerning the (none)-correlation of words, first, for both sequences, we verified that the number of words that contain n 1-bits over 128 obeys to a binomial law. Secondly, we checked that any generated word does not appear twice in the $1.5*10^6$ length sequence. This was an evidence for the LFSR-based sequence since $1.5*10^6$ is smaller than the LFSR cycle length (2^{128} -1) but we was not certain concerning the AES-based sequence.

In order to study the randomness of the sequence made up words that are joined end to end; we apply the whole statistical tests suite (NIST). The table 2 presents the test results.

The AES sequence passes all statistical tests in the test suite whereas type 2 LFSR passes only two tests. One explanation of this poor result concerning the type 2 LFSR is the low density of xor operations in the LFSR (in accordance to the chosen primitive polynomial). Thus two consecutive words have a similar structure. So the sequence made up words that are joined end to end is not random.

Table 2: Statistical test results

	AES	LFSR 2
1 : Freq	0.209694	0
2 : BlkFreq	0.471094	0.685858
3 : CuSum	0.228124	0
4 : Runs	0.613435	0
5 : LongRuns	0.600609	0
6 : Rank	0.556941	0.110155
7 : DFFT	0.417304	0
8 : Univ	0.475411	0
9 : Apen	0.805931	0
10 : Serial1	0.694507	0
11 : Serial2	0.289375	0
12 : LinComp	0.146473	0
13 : Aperiodic	0.565934	0
14 : Periodic	0.099107	0
15 : Random	0.614598	0

To sum up AES leads to better results than type 2 LFSR in term of randomness considering bitstreams sequences and 128-bits words sequences.

b) Sequences of 16 bits words:

Firstly, 16 bitstreams of the AES and the type 2 LFSR are studied. The proportion of bitstreams that pass a test is represented in figure 6 for the AES and figure 7 for the type 2 LFSR.

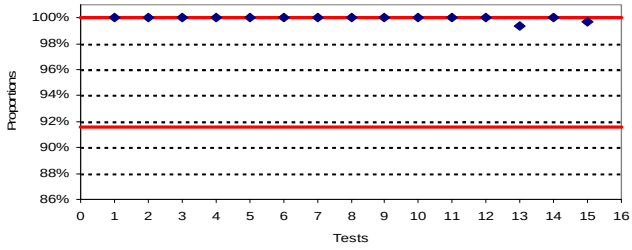

Fig. 6: percentage of AES bitstreams passing each test

All the proportions of AES bitstreams are included in the interval of confidence.

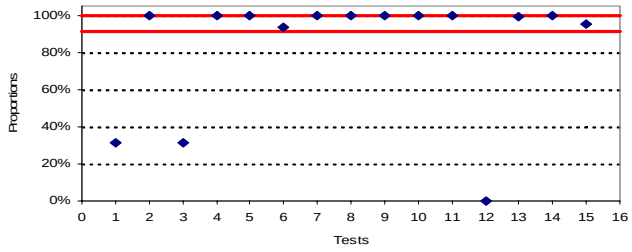

Fig. 7: percentage of type 2 LFSR bitstreams passing each test

The proportions of the 16 bitstreams that pass a test for the type 2 LFSR are similar to those obtained previously for the 128 bitstreams. Thus the conclusions for 128 bitstreams or for 16 bitstreams are similar; AES sequence is most random than the one obtained by mean of the type 2 LFSR.

In this experiment, a word is composed of 16 bits randomly selected from the 128 available output bits; the maximal number of combinations is 2^{16}. Note that in this case, a 1-bits word may appear several times in the $1.5*10^6$ sequence.

Regarding the randomness of the sequences composed of the 16-bits words, AES gives the better result. We performed the same experiments with other 16 bitstreams whose positions are randomly chosen among the 128 ones. The same results have been obtained. So, we guess the conclusion that the randomness of the sequence does not depend on the bit positions. This also means that no particular attention has to be paid when connecting a n-inputs CUT (with n<128) to the AES TPG.

As a conclusion of these experiments, the randomness properties of an AES are as good or even better as a type 2 LFSR. We thus consider the AES core as a good candidate for pseudo-random test pattern generation.

IV. FAULT SIMULATION

In this section, we evaluate the quality of the AES-based sequence as test sequence for sequential circuits. For that, several scanned versions of ISCAS'89 benchmark circuits are fault simulated using test sequences delivered either by the AES or by an LFSR. Fault simulation has been performed with the help of Synopsys Tetramax [11].

Figure 8 depicts how the AES core is connected to the circuits under test. When the number of scan chains is smaller than 128, some AES outputs are left unconnected.

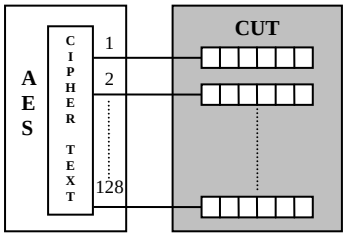

Fig. 8: test principle

The experiments have been done on three circuits with typical complexities (s9234 with 247 flip-flops, s13207 with 699 flip-flops and s38584 with 1464 flip-flops). For every benchmark, three scan configurations have been designed, namely 1, 16 and 128 scan chains. In the first configuration, the scan chain is connected to the LSB of the TPG. For the second configuration, the 16 scan chains are connected to outputs 2, 18, 25, 38, 40, 41, 59, 71, 75, 80, 98, 100, 101, 110, 111 and 125 of the TPG.

Tables 3, 4 and 5 report the stuck-at fault coverage for the three circuits. In these tables, the first column gives the number of scan patterns applied to the circuit. Column 2 indicates the scan configuration (1, 16 or 128 scan chains). Columns 3 to 5 give the resulting fault coverage for the three kinds of TPGs.

It can be seen that, whatever the circuit and whatever the scan configuration, the type 2 LFSR and the AES outperform the type 1 LFSR. This is in accordance with the results concerning randomness.

The results for the type 2 LFSR and the AES are of the same order. It appears that the type 2 LFSR gives better results when the size of the circuit is small or moderate and

when there is a single scan chain. In all other cases, the AES leads to higher fault coverage. This observation is explained by the results given in table 2.

Table 3: circuit s9234

Patterns	Number of scan chain	Type1 LFSR	Type 2 LFSR	AES
		FC (%)	FC (%)	FC (%)
42449	1	64.29	90.17	88.95
	16	62.98	88.96	88.96
	128	61.29	90.54	90.01
84898	1	64.29	91.26	91.17
	16	62.98	90.82	91.49
	128	61.29	91	91.15
127347	1	64.29	91.56	91.35
	16	62.98	91.07	92.43
	128	61.29	91.37	91.64
169796	1	64.29	92.01	91.98
	16	62.98	91.22	93.45
	128	61.29	91.53	92.74

Table 4: circuit s13207

Patterns	Number of scan chain	Type1 LFSR	Type 2 LFSR	AES
		FC (%)	FC (%)	FC (%)
15000	1	81.08	99.37	96
	16	78.69	94.62	95.37
	128	79.56	86.48	94.93
30000	1	81.08	99.38	98.21
	16	78.69	95.45	97.71
	128	79.56	86.75	97.14
45000	1	81.08	99.38	98.59
	16	78.69	95.83	98.57
	128	79.56	86.81	98.32
60000	1	81.08	99.38	99.06
	16	78.69	95.93	98.93
	128	79.56	86.84	98.93

Table 5: circuit s38584

Patterns	Number of scan chain	Type1 LFSR	Type 2 LFSR	AES
		FC (%)	FC (%)	FC (%)
7161	1	72.82	94.14	94.97
	16	82.39	93.54	94.21
	128	82.67	93.93	94.36
14322	1	72.82	95.31	95.59
	16	82.39	94.90	94.99
	128	82.67	94.97	95.60
21483	1	72.82	95.77	95.99
	16	82.39	95.54	95.48
	128	82.67	95.52	95.89
28644	1	72.82	95.93	96.15
	16	82.39	95.81	96.19
	128	82.67	96.05	96.20

Test time being directly linked to the scan chain length, increasing the number of scan chains is the most common approach to reduce test time. Thus, the AES is an interesting alternative to LFSR as TPG for nowadays circuits of large complexity.

V. Discussion

Concerning implementation issues, it must be noted first that the test controller is similar for an AES generator than for a LFSR generator. Furthermore, concerning random resistant faults, it is generally necessary to apply specific test patterns. Similarly to the LFSR reseeding technique, the proposed TPG architecture allows to apply pre-computed test patterns with the same extra hardware.

Concerning test time, as presented in this paper, the AES produces one test vector after every encryption cycle. Because of implementation issues, one encryption cycle requires at least 10 clock cycles (one encryption round per clock cycle). Conversely, a LFSR-based test pattern generator is able to produce one test vector every clock cycle, which gives an advantage to LFSR with regard to test time. A solution based on using the result of an encryption round instead of an encryption cycle is currently under study. It would allow obtaining the same test time for both the AES and the LFSR solutions.

VI. Conclusion

In this paper, we have shown that an AES cryptographic core could be used as pseudo-random test pattern generator. In the context of secure circuits, this allows to avoid implementing specific extra hardware dedicated to test generation.

First, the randomness of generated sequences has been empirically questioned thanks to the NIST test suite. The results show that AES-based sequences exhibit better randomness characteristics than those generated from LFSRs.

Furthermore, these sequences have been applied to three benchmark circuits, with three scan configurations for each. The results in terms of faults coverage show that AES is a good TPG for circuit with numerous scan chains. This corroborates the first point. AES-based BIST is a valuable alternative to standard random LFSR-based BIST.

Refences

[1] B. Yang, K. Wu, R. Karri, "Scan-based Side-Channel Attack on Dedicated Hardware Implementations on Data Encryption Standard", Proc. International Test Conference (ITC 2004), pp 339-344

[2] B.Yang, K.Wu, R.Karri, "Secure Scan: A Design-for-Test Architecture for Crypto Chips", Proc. Design Automation Conference 2005 p135-140

[3] FIPS 197, Advanced Encryption Standard (AES), November 2001.

[4] B. Schneier, "Applied cryptography", John Wiley & Sons Ed., 1996.

[5] A. Schubert, W. Anheier, "On random pattern testability of Cryptographic VLSI cores", ETW 99

[6] B. Yang, R. Karri, "Crypto BIST: A Built-In Self Test Architecture for Crypto Chips", 2nd Workshop on fault diagnosis and tolerance in cryptography (FDTC 2005), pp 95-108

[7] "A statistical test suite for random and pseudorandom number generators for cryptographic applications" NIST Special Publication 800-22 (with revisions dated May 15, 2001)

[8] J. Soto, L. Bassham, "Randomness Testing of the Advanced Encryption Standard Finalist Candidates1", National Institute of Standards and Technology, [2000]

[9] P. Hellekalek, S. Wegenkittl, "Empirical evidence concerning AES", ACM Trans. Model. Comput. Simul., Vol. 13, Issue 4 (Oct. 2003), 322-333

[10] M. Abramovici, M. A. Breuer and A. D. Friedman, "Digital Systems Testing and Testable Design", Computer Science Press

[11] www.synopsys.com/products/test/tetramax_ds.html