International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 23 (2015) pp 43866-43872
© Research India Publications. http://www.ripublication.com

Power Optimized LBIST for Cryptographic Applications

N. Rama Priya
M. Tech Student, ECE (DECS) Gudlavalleru Engineering College Gudlavalleru, Krishna Dist., AP, India-521356
e-mail: ramapriya42@gmail.com

A. V. N. Tilak
Professor of ECE, Dean-P. G. Studies and R&D Gudlavalleru Engineering College
Gudlavalleru, Krishna Dist., AP, India-521356 e-mail: avntilak@yahoo.com
Telephone: 08674-273737, Fax No: 08674-273957

Abstract –
Hardware implementations are prone to security problems due to a random fault or a deliberate attack. The traditional testing methods are good at detecting random faults, but they do not provide adequate protection. Cryptographic methods are used to protect confidential information against unauthorized modification or disclosure. In this paper a Logic Built-In Self-Test (LBIST) for Linear Feedback Shift Register (LFSR) based cryptographic systems, is presented which allows to cover maximum percentage of single stuck-at faults with low power and less time. Here a crypto device with low complexity and high security is designed by using Advanced Encryption Standard (AES) algorithm along with LBIST technique. This is used to secure the data by making use of symmetric key based authentication. Further Scan Chain Reordering Algorithm (SCRA) provides a reduction in power consumption by 57% while clock optimization technique reduces accessing and operational times of crypto operations by 9%.

Key Words: LBIST, LFSR, crypto device, AES, symmetric key, power optimization.

I. Introduction

Today technology is moving towards deep submicron level as the number of devices and applications which send and receive data are increasing rapidly and also the data transfer rates are becoming higher. In many applications, this data requires a secured connection which is usually achieved by cryptography. In the past several decades cryptography has become an indispensable tool for realizing secure communications [1]. Confidentiality is ensured through cryptographic mechanisms, generally implemented on co-processors. Nowadays, encryption is emerging as a disintegrable part of all communication networks and information processing systems, for protecting both stored and in-transit data world. Secure circuits are commonly used for applications such as e-banking, paytm, cell phone conversations, military applications, to maintain financial and legal files, medical reports, etc., because they hold personal data and must process secure operations. Security requirements such as source/sink authentication, data integrity, confidentiality, or tamper resistance are maintained by means of several dedicated components.

A hardware fault can compromise the security of a cryptographic system. To make possible periodic fault detection in functional circuits during their lifetime, cryptographic systems often employ Logic Built-In Self-Test. Feedback Shift Register (FSR) [2] based cryptographic systems are the fastest and the most power-efficient cryptographic systems for hardware applications.

Many cryptographic algorithms [3] were proposed, such as the Data Encryption Standard (DES), the Elliptic Curve Cryptography (ECC), the Advanced Encryption Standard (AES). Here the implementation of AES algorithm with LBIST is presented to increase the data transfer speed and security. The power optimization for the circuit is attained through scan chain reordering and clock optimization technique used in the circuit results in less time.

II. Background

In 1997 the National Institute of Standards and Technology (NIST), a branch of the US government, started a process to identify a replacement for the Data Encryption Standard (DES). It was generally recognized that DES was not secure because of advances in computer processing power. The goal of NIST was to define a replacement for DES that could be used for non-military information security applications by US government agencies. It was recognized that commercial and other non-government users would benefit from the work of NIST and that the work would be generally adopted as a commercial standard.

III. Proposed Work

In this paper the design of Logic Built-In Self-Test (LBIST) for Linear Feedback Shift Register (LFSR) based cryptographic system (AES crypto core) is implemented. The block diagram of the proposed system is shown in figure 1.

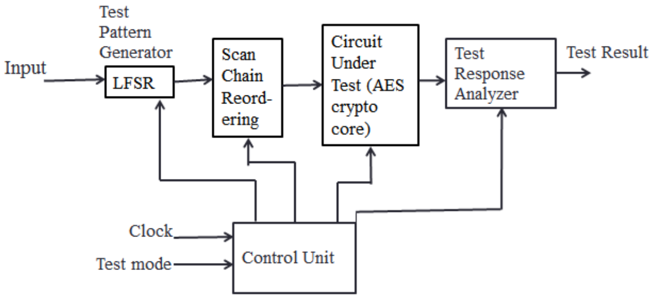
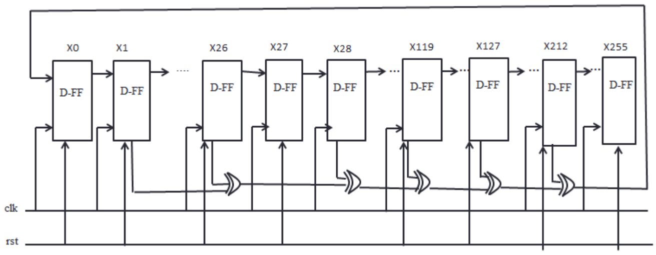


Figure 1: Block diagram of LBIST for AES crypto core

3.1 LFSR

Linear Feedback Shift Registers are used to generate all test patterns in a sequence depending upon a) the initial input sequence and b) the tap connections [4]. Boolean functions used in cryptographic systems are commonly represented in Algebraic Normal Form (ANF) [5] which is represented by f_{255}. Here the ANF and its LFSR structure which is designed to generate all the 256-bit input combination patterns is shown in figure 2.



$$f_{255} = x1 \oplus x26 \oplus x28 \oplus x119 \oplus x127 \oplus x212$$

Figure 2: 256-bit linear feedback shift register

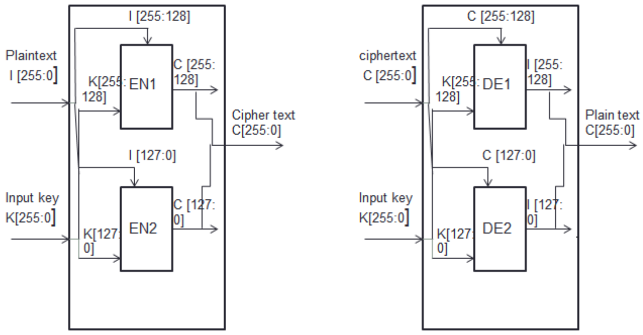
An LFSR is a shift register that, when clocked, advances the signal through the register from one bit to the next most-significant bit. Some of the outputs are combined in exclusive-OR configuration to form a feedback mechanism. Linear feedback shift registers make extremely good pseudorandom pattern generators. When the outputs of the flip-flops are loaded with a seed value (anything except all 0s, which would cause the LFSR to produce all 0 patterns) and when the LFSR is clocked, it will generate a pseudorandom pattern of 1s and 0s. A maximal-length LFSR produces the maximum number of pseudo random patterns possible and has a pattern count equal to $2^n - 1$, where n is the number of register elements in the LFSR.

3.2 AES Crypto Core

The AES crypto core is a symmetrical block cipher that consists of encryption, key generation and decryption processes. These are explained in the following subsections.

3.2.1 AES Encryption

Encryption is the process of converting information from readable to unreadable format. The message prior to the encryption process is called the plain text while the scrambled data after the encryption is called the cipher text. The plain text can be recovered from the cipher text with a decryption process using a key. The algorithm that can perform encryption and decryption is called a cipher.
The AES encryption algorithm is a block cipher that uses an encryption key and several rounds of encryption [6]. The AES ciphers a block of 256 bits plain text into a 256 bits cipher text with the help of a 256 bits secret key. The 256 bits plaintext is organized into two 4*4 matrices of 16 bytes. The data flow for input of length 256 bits is as shown in figure 3. After a first XOR operation between key and the plaintext, the algorithm consists in several rounds: 10, 12 or 14 rounds according to the key length 128, 192 or 256 bits. Every round except the last one is composed of four operations. The structure of AES encryption is shown in figure 4.



3(a)Encryption block 3(b)Decryption block

Figure 3: Basic block diagram for data flow of AES algorithm for input of length 256 bits.

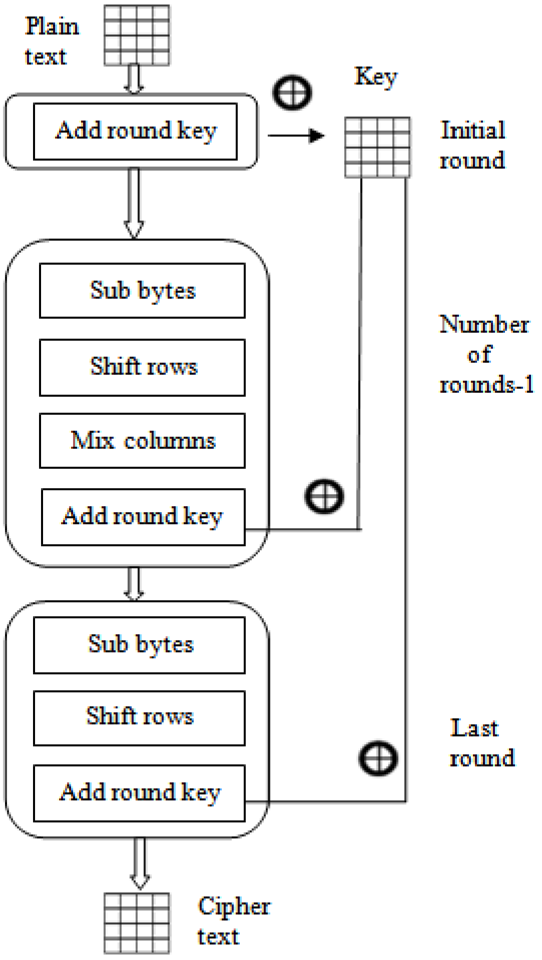
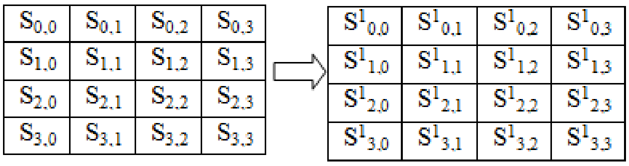


Figure 4: AES encryption structure

1) ***Sub bytes:*** Sub bytes uses S-Box lookup table that is comprise with 256 entries of 8-bit data. That is substituted to in place of the state values as shown in figure 5.

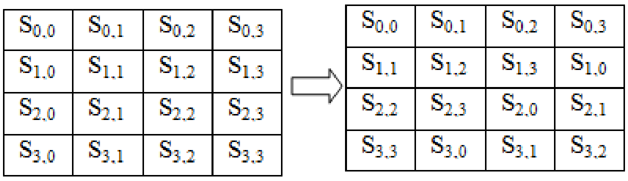


$S_{0.0}$	$S_{0.1}$	$S_{0.2}$	$S_{0.3}$
$S_{1.0}$	$S_{1.1}$	$S_{1.2}$	$S_{1.3}$
$S_{2.0}$	$S_{2.1}$	$S_{2.2}$	$S_{2.3}$
$S_{3.0}$	$S_{3.1}$	$S_{3.2}$	$S_{3.3}$

$S^1_{0.0}$	$S^1_{0.1}$	$S^1_{0.2}$	$S^1_{0.3}$
$S^1_{1.0}$	$S^1_{1.1}$	$S^1_{1.2}$	$S^1_{1.3}$
$S^1_{2.0}$	$S^1_{2.1}$	$S^1_{2.2}$	$S^1_{2.3}$
$S^1_{3.0}$	$S^1_{3.1}$	$S^1_{3.2}$	$S^1_{3.3}$

Figure 5: Sub bytes operation

2) ***Shift rows:*** Shift rows consists of left circular shifts on the matrix line. As shown in figure 6 the zeroth row is not shifted at all. First row shifts cyclically one byte to the left, second row shifts two bytes and third row shifts three bytes to the left.



$S_{0.0}$	$S_{0.1}$	$S_{0.2}$	$S_{0.3}$
$S_{1.0}$	$S_{1.1}$	$S_{1.2}$	$S_{1.3}$
$S_{2.0}$	$S_{2.1}$	$S_{2.2}$	$S_{2.3}$
$S_{3.0}$	$S_{3.1}$	$S_{3.2}$	$S_{3.3}$

$S_{0.0}$	$S_{0.1}$	$S_{0.2}$	$S_{0.3}$
$S_{1.1}$	$S_{1.2}$	$S_{1.3}$	$S_{1.0}$
$S_{2.2}$	$S_{2.3}$	$S_{2.0}$	$S_{2.1}$
$S_{3.3}$	$S_{3.0}$	$S_{3.1}$	$S_{3.2}$

Figure 6: Shift rows transformation

3) ***Mix columns:*** Mix columns as shown in figure 7 is a multiplication of state columns by a fixed matrix in the Galois Field GF(2^8), given by the polynomial a(x)= {03}x^3+{01}x^2+{01}x+{02}.

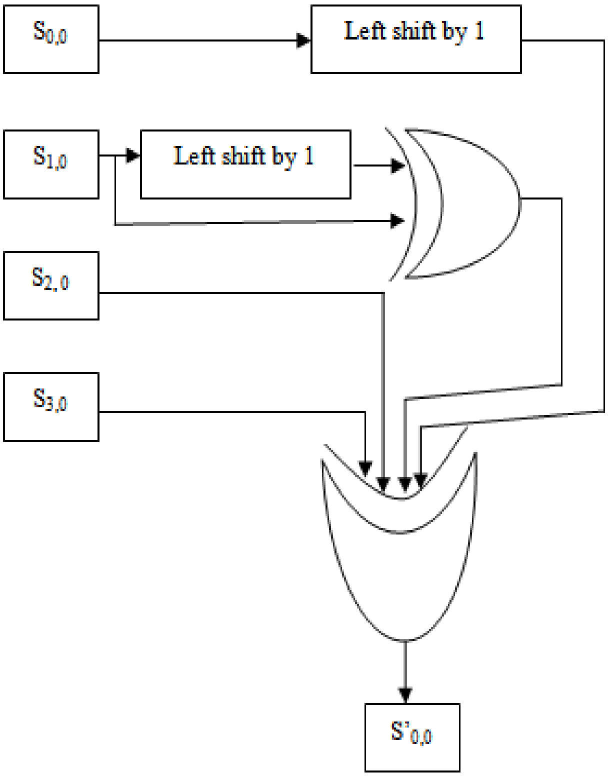


Figure 7: Matrix multiplication transformation

4) *Add round key:* Add round key is a XOR operation between the partially ciphered text and the round key. Round key being derived from the initial secret key. The last round does not execute the mix columns operation. Actually for 128-bit key size 10 rounds of operations takes place but here the key of length 256-bit is taken as the algorithm is implemented on 256-bit input data by dividing into two 128-bit blocks which runs individually as shown in figure 3. So, without loss of generality, we assume here-after, overall 10 rounds of operations for 256-bit key. Figure 4 presents the basic iterative operation of the AES encryption algorithm.

3.2.2 AES Key Generation

The key expansion term is used to describe the operation of generating all round keys from the original input key. The initial round key will be the original key in case of encryption and the last group of the generated expansion keys in case of decryption [6]. The Round CONstant (RCON) value changes for each individual round. The whole operation is shown in figure 8.

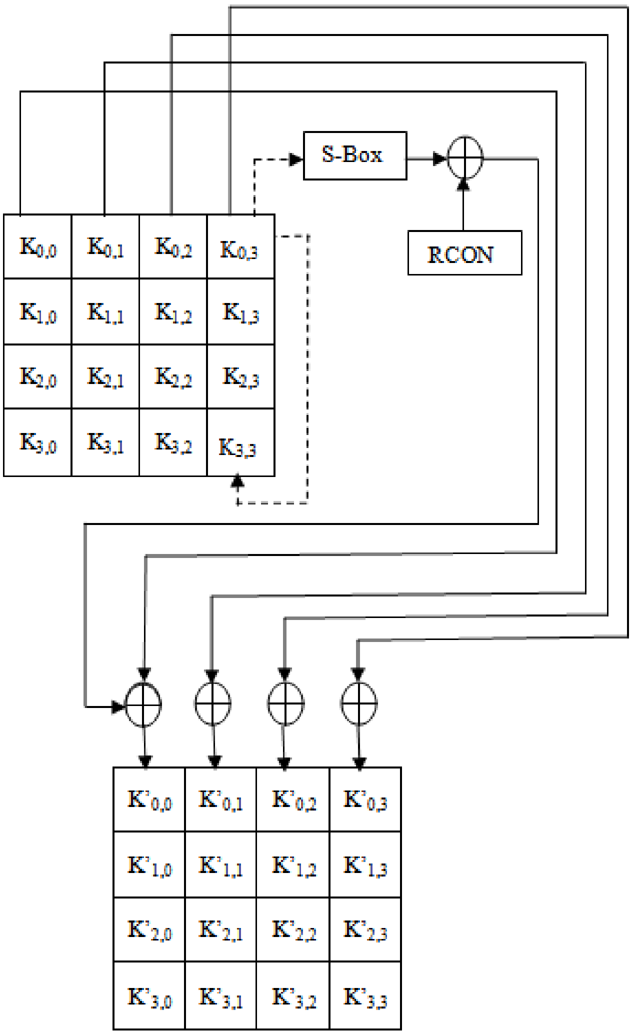

S-Box
RCON
$K_{0,0}$ $K_{0,1}$ $K_{0,2}$ $K_{0,3}$
$K_{1,0}$ $K_{1,1}$ $K_{1,2}$ $K_{1,3}$
$K_{2,0}$ $K_{2,1}$ $K_{2,2}$ $K_{2,3}$
$K_{3,0}$ $K_{3,1}$ $K_{3,2}$ $K_{3,3}$
$K'_{0,0}$ $K'_{0,1}$ $K'_{0,2}$ $K'_{0,3}$
$K'_{1,0}$ $K'_{1,1}$ $K'_{1,2}$ $K'_{1,3}$
$K'_{2,0}$ $K'_{2,1}$ $K'_{2,2}$ $K'_{2,3}$
$K'_{3,0}$ $K'_{3,1}$ $K'_{3,2}$ $K'_{3,3}$

Figure 8: AES key generation structure

3.2.3 AES Decryption

AES decryption is inverse operation of the encryption. AES decryption process takes cipher text as its input. As shown in figure 9, the state vector goes through mathematical transformation and the resultant is the plain text which was given as input to the AES encryption.

The process is nearly identical to the encryption counterpart, except that the rotation is towards right and the Inverse S-Box is used for substitution and for the Inverse mix columns operation, the difference is the multiplication matrix, given by the polynomial [7] $a^{-1}(x)$ = {0b} x3+ {0d} x2+ {09} x+ {0e} which is inverse of the matrix mentioned in encryption. In the last round, Inverse mix column transformation is not done and the output of Inverse add round key of last stage is taken as the decrypted data.

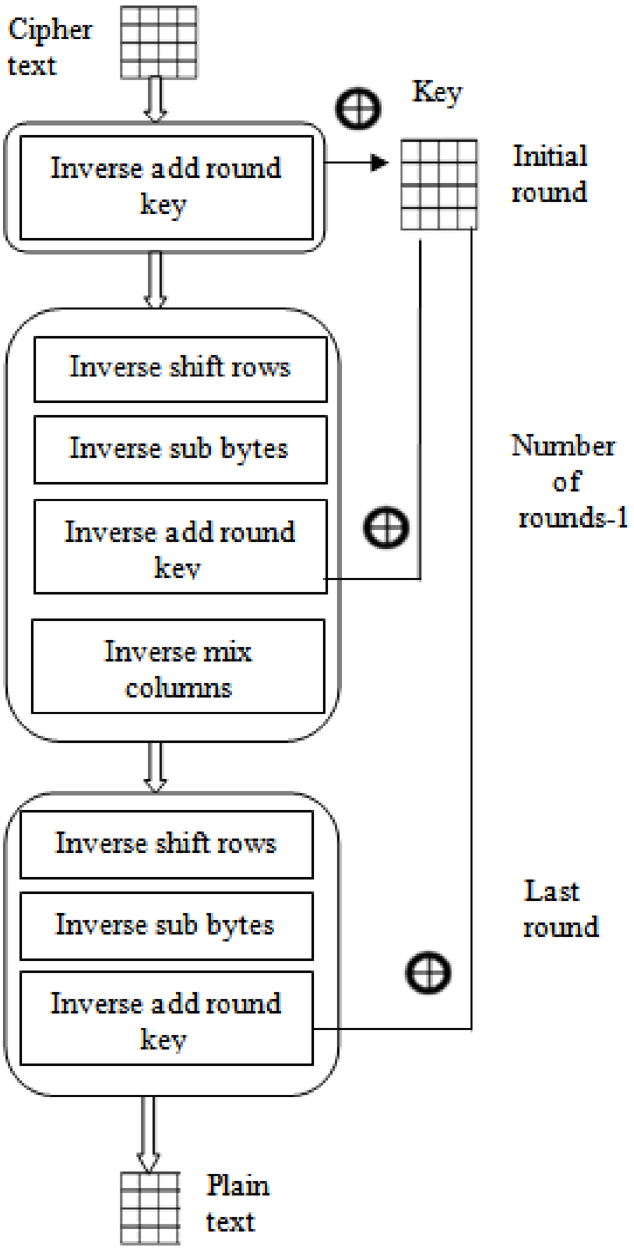


Figure 9: AES decryption structure

3.3 LBIST

Built-in self-test for logic circuits or logic BIST (LBIST) is an effective solution for the test cost, test quality, and test reuse problems. Logic BIST implements most Automatic Test Generation (ATG) functions on chip so that the test cost can be reduced through less test time, less tester memory requirement, or a cheaper tester. Logic BIST [8] applies a large number of test patterns so that more defects, either modelled or un-modelled, can be detected. Furthermore, a BISTed-core makes SoC testing easier. There are two main functions that must be performed on-chip in order to implement BIST: test pattern generation and output response analysis. The most common BIST schemes are based on pseudorandom test patterns generation using linear feedback shift registers and output response compaction using analysers. The signature is compared with golden signature to check test pass/fail information that indicates whether defect or not. The implementation of LBIST is shown in figure10.

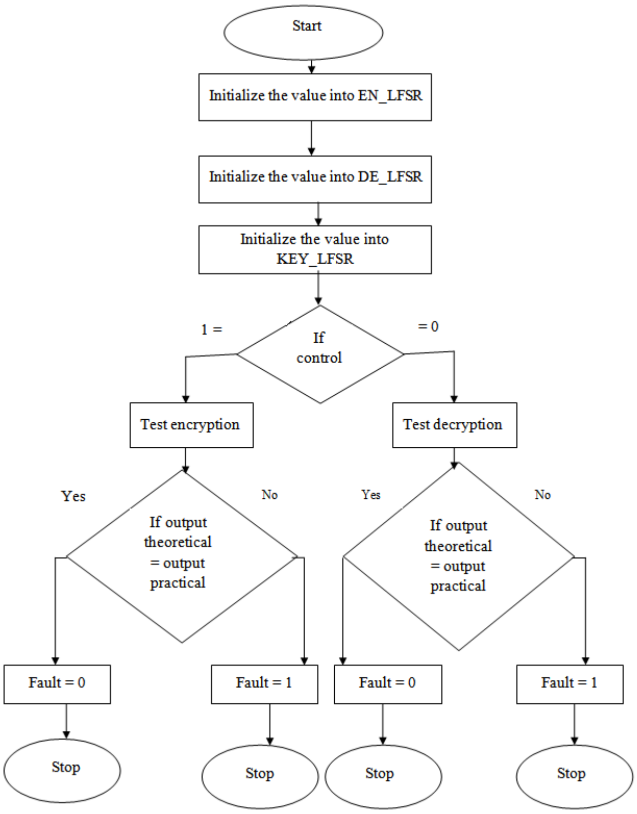
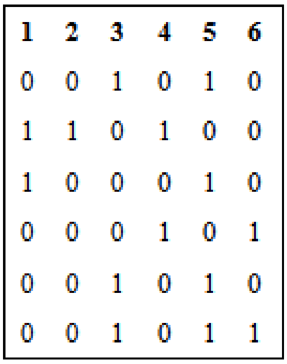

Start
Initialize the value into EN_LFSR
Initialize the value into DE_LFSR
Initialize the value into KEY_LFSR
1 =
If control
= 0
Test encryption
Test decryption
Yes
No
If output theoretical = output practical
Yes
No
If output theoretical = output practical
Fault = 0
Fault = 1
Fault = 0
Fault = 1
Stop
Stop
Stop
Stop

Figure 10: Implementation of LBIST

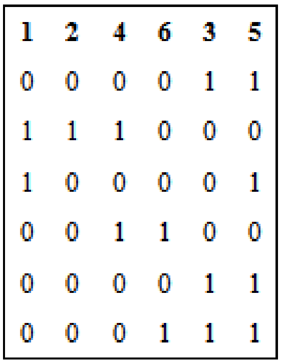
3.4 Power Optimization

Here scan-chain reordering algorithm [9] is implemented for power optimization. In this technique some cells of the ordered scan chain will be reordered again in order to reduce the peak power which may result during the test cycle. Scan reordering is an effective technique for saving scan power during testing since it requires no extra Design For Testability (DFT) logic and does not affect the fault coverage or test application time. For example consider the table 1 which represents the original test set and table 2 represents reordered test set and it is observed that in original test set vertical transition are 20 and after reordering, it decreases to 8 transitions.

Table 1:Original test set Table 2:Reordered test set



1	2	3	4	5	6
0	0	1	0	1	0
1	1	0	1	0	0
1	0	0	0	1	0
0	0	0	1	0	1
0	0	1	0	1	0
0	0	1	0	1	1

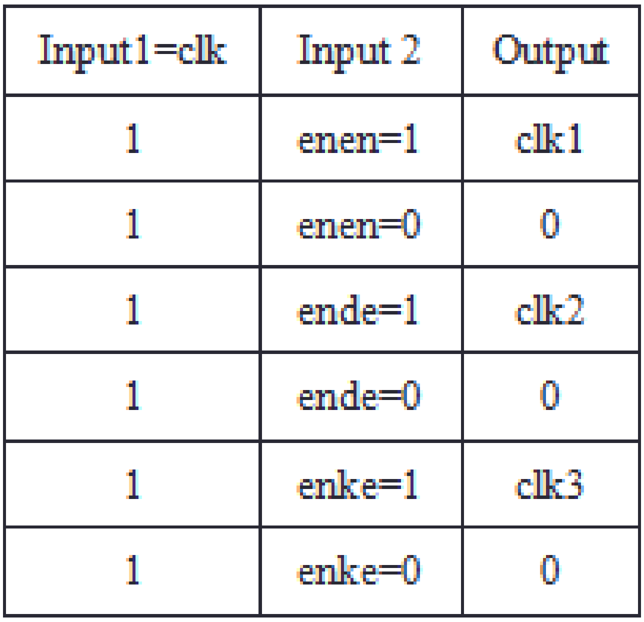


1	2	4	6	3	5
0	0	0	0	1	1
1	1	1	0	0	0
1	0	0	0	0	1
0	0	1	1	0	0
0	0	0	0	1	1
0	0	0	1	1	1

3.5 Timing Optimization

Further a clock optimization technique[10] is employed which reduces the accessing and operational times. This clock skew technique enables the clock only to the required operations at current time and bypasses the clock for those operations which are not intended to. Here in AES algorithm when control is set to 1 the encryption phase runs where the decryption is idle so there the clock gets bypassed for decryption phase operations. So here the operational time gets reduced and during s-box operations where conversion of integer operations takes place by accessing the memory there the accessing time gets reduced. The implementation of clock optimization logic is shown in table 3 by considering clk1 to enable encryption(enen), clk2 to enable key(enke), clk3 to enable decryption(ende) and clk is the main clock which is always 1.

Table 3:Implementation of clock optimization logic



Input1=clk	Input 2	Output
1	enen=1	clk1
1	enen=0	0
1	ende=1	clk2
1	ende=0	0
1	enke=1	clk3
1	enke=0	0

IV. Results

The circuit is implemented by using VHDL as the hardware description language. The software used for the simulation of the circuit is Xilinx ISE 12.1.

Figure 11 shows the simulation results of AES encryption phase i.e. when control=1 encryption phase runs. Here the ‘input_encryption’ and ‘input_key’ both of length 256 bits are taken as inputs and gives ‘output_encryption’ as output which is a cipher text of length 256 bits.

Figure 12 shows the simulation results of AES decryption phase i.e. when control=0 decryption phase runs. Here the ‘input_decryption’(given the value of ‘output_encryption’ from figure 11 which is the cipher text) and ‘input_key’ both of length 256 bits are taken as inputs and gives ‘output_decryption’ which is a plain text of length 256 bits as output. From figure 11 ‘input_encryption’ signal = ‘output_decryption’ signal, in figure 12 which indicates that the decryption output is same as given encryption input.

In figure 13 the simulation results of BIST are shown when there is no fault i.e. the self-test is performed and comparision between the output of theoretical circuit and practical circuit is done and indicates as 1 if fault is present or else indicates as 0 if there is no fault. Here ‘w5_enout’, ‘w6_deout’ shows the theoretical circuit output of encryption and decryption phases. The practical circuit output is given by ‘w8_enout_f’, w9_deout_f’ for encryption and decryption phases respectively. As the practical circuit is fault-free(i.e. (‘w5_enout’ = ‘w8_enout_f’) and (‘w6_deout’ = ‘w9_deout_f’)), the ‘faulten’(indication of fault at encryption side) and ‘faultde’(indication of fault at decryption side) are both 0 indicating that the circuit is fault-free.

Figure 14 shows the simulation results of BIST when there is fault at encryption phase(i.e. (‘w5_enout’ ≠ ‘w8_enout_f’) and (‘w6_deout’ = ‘w9_deout_f’)). The ‘faulten’ signal=1(indication of fault at encryption side) and ‘faultde’ signal=0(indication of fault at decryption side) indicating that the circuit is faulty at encryption side. Similarly if there is no fault at encryption, ‘faulten’ signal will be 0 and if there is fault at decryption ‘faultde’ signal will be 1. If there is fault at both encryption and decryption phases both ‘faulten’ and ‘faultde’ signals will be 1. Here the practical circuit is taken as another theoretical circuit with inducing some fault.

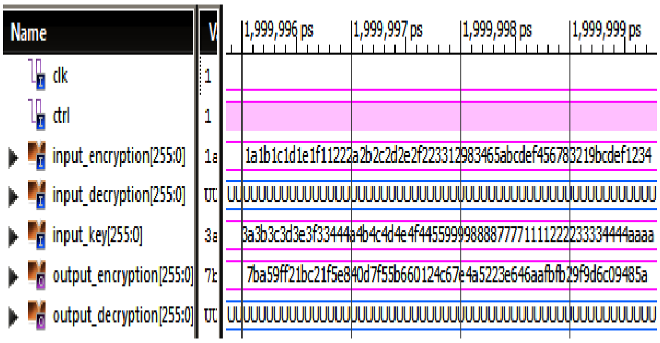
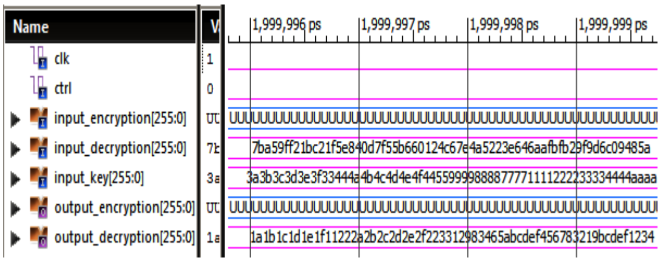
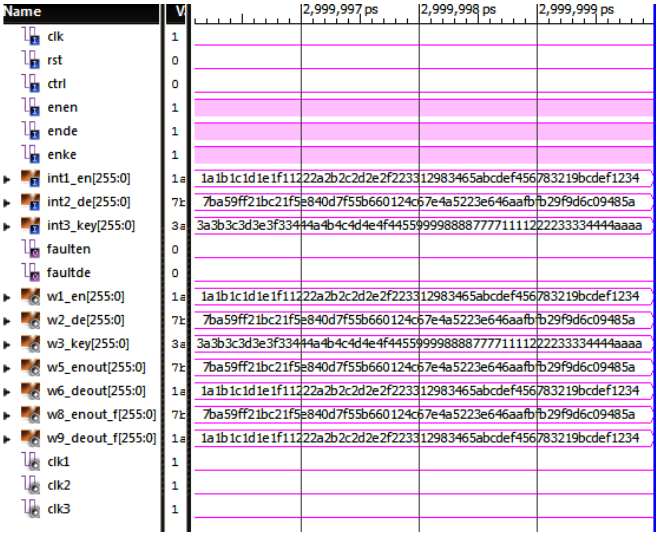

Name
V
1,999,996 ps
1,999,997 ps
1,999,998 ps
1,999,999 ps
clk
1
ctrl
1
input_encryption[255:0]
1a
1a1b1c1d1e1f11222a2b2c2d2e2f22331983465abcdef45678321 9bcdef1234
input_decryption[255:0]
UU
UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU
input_key[255:0]
3a
3a3b3c3d3e3f33444a4b4c4d4e4f4455999988887777111122223333444aaaa
output_encryption[255:0]
7b
7ba59ff21bc21f5e840d7f55b660124c67e4a5223e646aafbfb29f9d6c09485a
output_decryption[255:0]
UU
UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU

Figure 11: Simulation results of AES encryption phase in functional mode (when control=1)



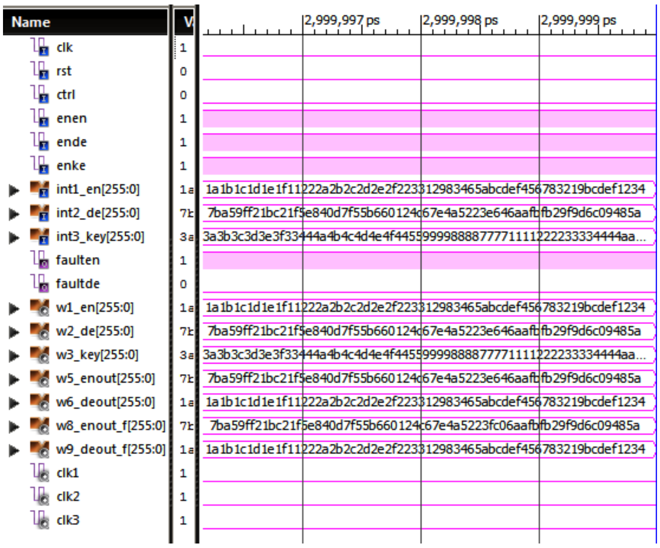
Name	V	
clk	1	
ctrl	0	
input_encryption[255:0]	UU	UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU
input_decryption[255:0]	7b	7ba59ff21bc21f5e840d7f55b660124c67e4a5223e646aafbfb29f9d6c09485a
input_key[255:0]	3a	3a3b3c3d3e3f33444a4b4c4d4e4f44559999888877771111222233334444aaaa
output_encryption[255:0]	UU	UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU
output_decryption[255:0]	1a	1a1b1c1d1e1f11222a2b2c2d2e2f223312983465abcdef456783219bcdef1234

Figure 12: Simulation results of AES decryption phase in functional mode (when control=0)



Name	V	
clk	1	
rst	0	
ctrl	0	
enen	1	
ende	1	
enke	1	
int1_en[255:0]	1a	1a1b1c1d1e1f11222a2b2c2d2e2f223312983465abcdef456783219bcdef1234
int2_de[255:0]	7b	7ba59ff21bc21f5e840d7f55b660124c67e4a5223e646aafbfb29f9d6c09485a
int3_key[255:0]	3a	3a3b3c3d3e3f33444a4b4c4d4e4f44559999888877771111222233334444aaaa
faulten	0	
faultde	0	
w1_en[255:0]	1a	1a1b1c1d1e1f11222a2b2c2d2e2f223312983465abcdef456783219bcdef1234
w2_de[255:0]	7b	7ba59ff21bc21f5e840d7f55b660124c67e4a5223e646aafbfb29f9d6c09485a
w3_key[255:0]	3a	3a3b3c3d3e3f33444a4b4c4d4e4f44559999888877771111222233334444aaaa
w5_enout[255:0]	7b	7ba59ff21bc21f5e840d7f55b660124c67e4a5223e646aafbfb29f9d6c09485a
w6_deout[255:0]	1a	1a1b1c1d1e1f11222a2b2c2d2e2f223312983465abcdef456783219bcdef1234
w8_enout_f[255:0]	7b	7ba59ff21bc21f5e840d7f55b660124c67e4a5223e646aafbfb29f9d6c09485a
w9_deout_f[255:0]	1a	1a1b1c1d1e1f11222a2b2c2d2e2f223312983465abcdef456783219bcdef1234
clk1	1	
clk2	1	
clk3	1	

Figure 13: Simulation results of LBIST mode without fault (when fault=0)

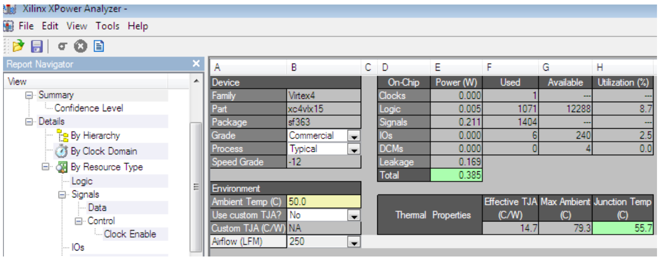


Name	V	
clk	1	
rst	0	
ctrl	0	
enen	1	
ende	1	
enke	1	
int1_en[255:0]	1a	1a1b1c1d1e1f11222a2b2c2d2e2f223312983465abcdef456783219bcdef1234
int2_de[255:0]	7b	7ba59ff21bc21f5e840d7f55b660124c67e4a5223e646aafbfb29f9d6c09485a
int3_key[255:0]	3a	3a3b3c3d3e3f33444a4b4c4d4e4f44559999888877771111222233334444aa...
faulten	1	
faultde	0	
w1_en[255:0]	1a	1a1b1c1d1e1f11222a2b2c2d2e2f223312983465abcdef456783219bcdef1234
w2_de[255:0]	7b	7ba59ff21bc21f5e840d7f55b660124c67e4a5223e646aafbfb29f9d6c09485a
w3_key[255:0]	3a	3a3b3c3d3e3f33444a4b4c4d4e4f44559999888877771111222233334444aa...
w5_enout[255:0]	7b	7ba59ff21bc21f5e840d7f55b660124c67e4a5223e646aafbfb29f9d6c09485a
w6_deout[255:0]	1a	1a1b1c1d1e1f11222a2b2c2d2e2f223312983465abcdef456783219bcdef1234
w8_enout_f[255:0]	7b	7ba59ff21bc21f5e840d7f55b660124c67e4a5223fc06aafbfb29f9d6c09485a
w9_deout_f[255:0]	1a	1a1b1c1d1e1f11222a2b2c2d2e2f223312983465abcdef456783219bcdef1234
clk1	1	
clk2	1	
clk3	1	

Figure 14: Simulation results of LBIST mode with fault at encryption phase (when fault=1)

Power Analysis

Power Analysis is done using Xilinx power analyzer. Power reduction is attained by using scan chain reordering technique. Without using scan chain reordering the power consumption for the circuit is 0.385W as shown in figure 15 and with using scan chain reordering the power consumption for the circuit is 0.165W as shown in figure 16.



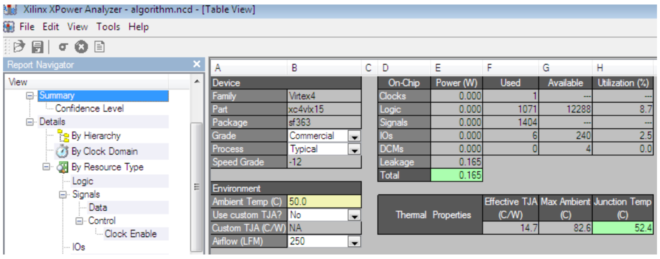
Device	
Family	Virtex4
Part	xc4vlx15
Package	sf363
Grade	Commercial
Process	Typical
Speed Grade	-12

Environment	
Ambient Temp (C)	50.0
Use custom TJA?	No
Custom TJA (C/W)	NA
Airflow (LFM)	250

On-Chip	Power (W)	Used	Available	Utilization (%)
Clocks	0.000	1	---	---
Logic	0.005	1071	12288	8.7
Signals	0.211	1404	---	---
IOs	0.000	6	240	2.5
DCMs	0.000	0	4	0.0
Leakage	0.169			
Total	0.385			

Thermal Properties	Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)
	14.7	79.3	55.7

Figure 15: Power report without using scan chain reordering algorithm



Device	
Family	Virtex4
Part	xc4vlx15
Package	sf363
Grade	Commercial
Process	Typical
Speed Grade	-12

Environment	
Ambient Temp (C)	50.0
Use custom TJA?	No
Custom TJA (C/W)	NA
Airflow (LFM)	250

On-Chip	Power (W)	Used	Available	Utilization (%)
Clocks	0.000	1	---	---
Logic	0.000	1071	12288	8.7
Signals	0.000	1404	---	---
IOs	0.000	6	240	2.5
DCMs	0.000	0	4	0.0
Leakage	0.165			
Total	0.165			

Thermal Properties	Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)
	14.7	82.6	52.4

Figure 16: Power report using scan chain reordering algorithm

Timing Analysis

The time consumption is reduced using clock optimization technique. Without using clock optimization technique maximum output required time after clock is 4.142ns and with using clock optimization technique maximum output required time after clock is 3.779ns.

V. Conclusion

In the context of secure circuits, LBIST approaches appear as good alternatives since they do not rely on visible scan chains. The logic BIST is highly efficient with a high fault coverage and the best part of the scheme is that it's speed which reduces the interface to the tester, tester memory and tester time. In this work, it is presented using an AES-based cryptographic core commonly embedded in secure systems. Generally only one AES core may be originally embedded in the system, it will be interesting to study concurrent test pattern generation and response analysis. Finally in this paper the implementation of LBIST for LFSR based cryptographic system is presented with reduction of power by 57% using scan chain reordering technique and reduction of time by 9% using clock optimization technique.

VI. References

[1] Bart Preneel, "Cryptography for Network Security: Failures, Successes and Challenges,"Katholieke Universiteit Leuven and IBBT Dept. Electrical

Engineering ESAT/COSIC, Kasteelpark Arenberg 10 Bus 2446, B-3001 Leuven, Belgium, pp. 1-19, 2010.

[2] Elena Dubrova, Mats N¨aslund, GoranSelander, “Secure and Efficient LBIST for Feedback Shift Register-Based Cryptographic Systems”, IEEE Conference Publications, pp. 978-1-4799-3415-7/14, 7/14 2014.

[3] Zhangxi Tan Chuang, Lin Hao Yin, “Optimization and Benchmark of cryptographic algorithms on network processors”, Bo Li Hong Kong University of Science and Technology Published by the IEEE computer Society, IEEE, pp. 55-69, 2004.

[4] Jui-Chieh Lin, Sao-Jie Chen, and Yu Hen Hu,Cycle-Efficient,“LFSRImplementation on Word-Based Microarchitecture”, IEEE Transactions On Computers, Vol. 62, No. 4, pp. 832-838, April 2013.

[5] Qichun Wang, Jie Peng, Haibin Kan and Xiangyang Xue, “Constructions of Cryptographically Significant Boolean Functions Using Primitive Polynomials”, IEEE transactions on information theory, Vol. 56, No. 6, pp. 3048-3053, June 2010.

[6] Xinmiao Zhang and Keshab K. Parhi, “Implementation Approaches for the Advanced Encryption Standard Algorithm”,IEEE, pp. 25-46, 2002.

[7] W.Suntiamorntut1, W. Wittayapanpracha “The Study of GF(2^8) AES Encryption for Wireless FPGA Node”, CISME Vol.2 No.3, pp. 40-46, 2012.

[8] Kiran George, Chien-In Henry Chen, ”Logic Built-In Self-Test for Core-Based Designs on System-on-a-Chip”, I2MTC-IEEE International Instrumentation and Measurement Technology Conference Victoria, pp. 1-4244-1541-1/08, 2008.

[9] Nan-Cheng Lai, Sying-Jyan Wang, and Yu-HsuanFu,”Low Power BIST with Smoother and Scan chain Reorder”, IEEE Transactions On Computer-Aided Design Of Integrated Circuits and Systems, Vol. 25, No. 11, pp. 2586-2594, November 2006.

[10] L. Benini, P. Vuillod, A. Bogliolo and G. De Micheli,”Clock Skew Optimization for Peak Current Reduction”, Journal of VLSI Signal Processing 16, pp. 117–130, 1997.

Biographical Sketch

N. Rama Priya received the B.Tech degree in Electronics and Communication Engineering from Nimra Women’s College of Engineering, Vijayawada, India. She is pursuing M.Tech in Digital Electronics and Communication Systems from Gudlavalleru Engineering College, Gudlavalleru.

A. V. N. Tilak has obtained his B.E., M.Tech. and Ph.D. from MIT Manipal, IIT Kanpur, and IIT Madras respectively. His areas of interest are Microelectronics, Digital Design, and Low Power VLSI Design. Dr. Tilak is a member of IEEE, Fellow IETE, Fellow IE(I), and Life member of ISTE.