FPGA Modeling of Fault-Injection Attacks on Cryptographic Devices

Vamsi Krishna Kosuri*, Dr. Fazal Noorbasha**

*(Student of MTech -VLSI, Department of Electronics and Communication Engineering, KL University, Guntur, AP, India. Email: krishna.kosuri@gmail.com)

** (Associate Professor, Department of Electronics and Communication Engineering, KL University, Guntur, AP, India. Email: fazalnoorbasha@kluniversity.in)

Abstract

Cryptographic applications like ATM and other commercial electronics are more susceptible to side channel attacks especially for Fault-injection attack and there is a strong need to design a secure multiplier which is the fundamental and crucial element of all cryptographic devices at which an intruder attacks are possible. This paper presents a design and modeling technique with FPGA to detect Fault-injection attacks with more sophisticated high speed multiplier architecture.

Keywords - Cryptographic Applications, Fault-injection attacks, KINTEX7- FPGA, Secure Multipliers and Detection Model.

I. INTRODUCTION

Cryptographic applications like ATM and other commercial electronics are more susceptible to side-channel attacks [1, 12] in which the fault injection attacks are more crucial at today's Deep Sub Micron Technology. Hence the identification and modeling of these attacks from the intruder is critical and became a challenge for current trend of system designs. Therefore it is essential to design the systems which are strong counterparts to side channel attacks.

It is obvious that the multiplier is the fundamental block for Public Key and Private Key , Cryptographic devices are mainly suffers with various side channel attacks includes for timing analysis, power analysis and fault injection attacks. The first two may not with the intruder side as the current system design has entered Deep Sub Micron Technology which will not allow the intruder to attack with negligible power and high speed of operation.

Public-key cryptographic devices are vulnerable to fault-injection attacks where the intruder may attack by injecting faults in to the fundamental element of the system with various sources and this always forces the technology to encounter these attacks by designing the basic elements more and more secure as the Deep Sub Micron Technology progress further, in this regards a number of secure multiplier architectures have been proposed with numerous coding techniques. Various arithmetic coding techniques have been proposed in order to detect the bit errors caused by intruder attacks [2,3] these codes may fall in to the categories like Linear codes , non linear[4], codes and multilinear codes etc. linear codes provide protection only in opposition to primitive adversaries with flawed attack capabilities, on the other hand nonlinear codes provides protection against strong adversaries, but at the price of high area overhead (200% - 400%).The proposed plan is a novel error detection technique based on protection mechanism of a multiplier which is a basic building block of many public-key cryptographic devices which is under nonlinear code error detection.

Since the multiplier will plays a vital role in all major Cryptographic algorithms, and also as per the design concerns it occupies much more area so as to consumes more power, therefore the multiplier design parameters such as speed, area and power must count into account for reliability issues of the Cryptographic devices. An assortment of multiplier architectures are proposed for Cryptographic algorithms to guarantee the security of the devices, in this regards various error detecting techniques are also proposed as mentioned earlier some of them are, linear arithmetic codes for example, Parity codes, Hamming codes, AN-codes etc. Non linear arithmetic codes like, Robust Codes and Multi linear arithmetic codes, each category has their strengths and limitations, therefore a proper study and observation is required to select a suitable detection algorithm.

Generally multiplier occupies more space when physical design, is concerned and also as the increased bit size results more occupancy of chip area, the other factor which effects the efficiency of the multiplier is its partial products, consequently there is a strong reason that multiplier design, for applications like Cryptography is always crucial and hence design of multiplier with good tradeoff between area, power and speed is essential especially for Deep Sub Micron era [11, 13, 14]. In order to achieve this tradeoff the multiplier structure should be designed in such away that it produces the required product terms with less number of partial products and to increase the speed, this task this could be achieved by adding some encoding technique to the multiplier structures.

The present article is alienated into three major areas. They are, design of secure multiplier and extending its design abilities such that it can hold out the intruder attacks, detection of fault-injection attacks, and finally simulation and its analysis of the proposed scheme.

The rest of this paper is structured as follows. In section II, multiplier design with extended features is described. In section III, design and analysis of detection model of fault-injection attacks. In section IV, the simulation results with synthesis reports.

II. DESIGN OF MULTIPLIER

It is obvious that for integer multiplication Wallace tree is an efficient hardware implementation [5], of a digital circuit and an n-bit Wallace-tree multiplier is on the order of *O*(log(n)) in terms of logic gates.

The advantage of the Wallace tree is that there are only *O*(log(n)) reduction layers, and each layer has *O*(1) propagation delay. As making the partial products is *O*(1) and the final addition is *O*(log(n)), the multiplication is only *O*(log(n)), not much slower than addition (however, much more expensive in the gate count). Where as adding partial products with regular adders would require *O*(log(n$^{2)}$) time. As per complexity theoretic perspective, the Wallace tree multiplication algorithm belongs to class NC. These computational tasks only consider gate delay but not accounting with wire delays, which can also be very substantial. The Wallace tree can be also represented by a tree of 3/2 or 4/2 adders. It is some times combined with Booth Encoding best yield of required out put.

Booth's Encoded multiplication algorithm is a multiplication algorithm that multiplies two signed binary numbers in two's complement notation by examining adjacent pairs of bits of the *N*-bit multiplier R in signed two's complement representation, including an implicit bit below the least significant bit, $R_{-1} = 0$. For each bit R_i, for *i* running from 0 to *N*-1, the bits R_i and R_{i-1} are considered. Where these two bits are equal, the product accumulator *P* remains unchanged. Where $R_i = 0$ and $R_{i-1} = 1$, the multiplicand times 2^i is added to *P*; and where Ri = 1 and $R_{i-1} = 0$, the multiplicand times 2^i is subtracted from *P*. The final value of *P* is the signed product.

Generally the representation of the multiplicand and product are not specified as these are both also in two's complement notation, similar to this multiplication concept any number systems that supports the addition and subtraction will also works good. The sequence is generally proceeds from LSB to MSB, starting at *i* = 0; the multiplication by 2^i is then typically replaced by incremental shifting of the *P* accumulator to the right between steps. Lower order bits could be shifted out and there by computing relevant addition and subtraction just on the highest *N* bits of *P*.

The algorithm can be described as converting strings of 1's in the multiplier to a high-order +1 and a low-order –1 at the ends of the string and whenever string run through the MSB, there is no high-order +1, and the resultant effect is representation of the appropriate value with negation.

By utilizing both features of above mentioned concepts, we will consider the Booth encoded Wallace tree as a basic element of error detection unit for Cryptographic application.

Let he multiplier has M-bits P and N-bits Q as input and generate M X N -bits output R, then P and Q can be represented as $P = \sum_{k=0}^{M-1} P_k 2^k$ and $Q = \sum_{l=0}^{N-1} Q_l 2^l$ such that the output R can be interpreted as $R = P \text{ X } Q = \sum_{k=0}^{M-1} (\sum_{l=0}^{N-1} P_k\, Q_l 2^{k+l})$. The output R is computed by adding the partial product $P_k\ Q_l$ together. The simplest implementation requires an N-bit adder and take M cycles to generate the output. Another implementation of multiplier is so called adder array multipliers which achieve higher speed at the cost of larger hardware. Several other technologies have been developed to improve the speed and reduce the power consumption of multiplier. There are two widely used approaches: booth algorithm and Wallace tree compressor, Booth algorithm can do multiplication on both non-negative and negative operand by using 2's complement number. Moreover by reducing the partial product count, this model can lead to substantial reduction of delay and effective area.

Wallace tree is a procedure of summing partial product bits in parallel. The Wallace tree can reduce both the critical path and number of required adders. The basic idea is using a full adder as a 3-2 compressor to reduce the product matrix. Because of both advantages of above mentioned concepts the Booth encoded – Wallace tree multiplier is considered as a secure basic functional element for the detection of fault-injection attacks and its specifications are summarized in the following table.

TABLE 1
SUMMARY OF BASIC MULTIPLIER DESIGN

S.No	Content	Description
1	Algorithm	Booth encoding
2	Radix	8
3	Structure	Wallace Tree
4	Multiplier size	20 –bit
5	HDL	Verilog
6	Development tool	Xilinx 14.1

The 20-bit, Radix-8 Booth encoded -Wallace Tree multiplier is included as the key element of detecting the fault- injection attacks and its Verilog Description is modeled using Xilinx14.1.

The functional building block of the Booth encoded-Wallace tree multiplier is shown in the following figure with various stages.

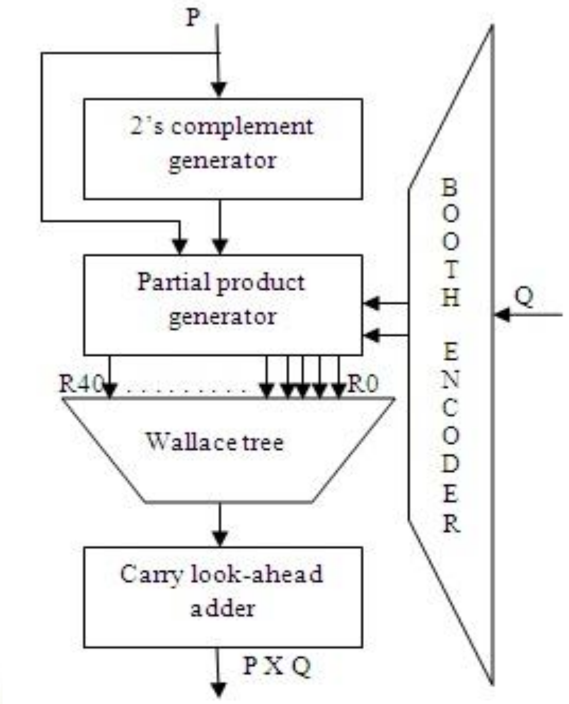

Fig.1. 20-Bit Booth Encoded Wallace Tree

III. MODELING OF FAULT DETECTION UNIT

Generally a fault can be injected at gate level with various sources, and it can be modeled is as shown in the figure.2 As mentioned earlier, numerous error detecting codes for Cryptographic applications are existed and their brief description is given here after.

Linear Arithmetic Codes: (ex: AN Codes, Hamming codes and Parity Codes etc.) they may work with a good accuracy but their abilities are limited because of the following reasons. They are suitable to a fastidious type of error (ex: error with odd multiplicity or byte error etc.), less sensitive to protect over unanticipated error, not recommendable to lazy channels, less and limited attack capability, not protective other than primitive adversities.

Non linear Arithmetic Codes: (ex: Robust Codes) [6,15], they are again divided into robust arithmetic residue codes and robust algebraic codes, these are best suitable to built Cryptographic devices which uses the AES (Advanced Encryption Standards) and recommendable especially devices uses arithmetic operations. They had advantages like: ability to overcome the weakness of Linear Codes, message dependability, best supportive for lazy channels, good fault detection capability and provision of equal protection over all error patterns. Still these codes are limited for practical implementation since, huge overhead of hardware because of need for encoding and decoding units.

Multi Linear Arithmetic Codes [7], the main principle of this category "is randomly selecting a code from multiple linear codes for each encoding and the corresponding decoding operations". These codes will provides the advantages like good error detection capability, less amount of hardware overhead, no need of non linear operations for encoding and decoding, negligible amount of bad errors and the best mates for lazy channels. There fore in order to describe the error attacker model we recommend Multi linear Arithmetic Codes as the prime principle along with Booth Encoded Wallace tree as a basic element of the operation. Further reduction of hardware overhead can be achieved with multi modulus multi linear arithmetic codes. Generally the fault injection can be represented as shown in the following figure.

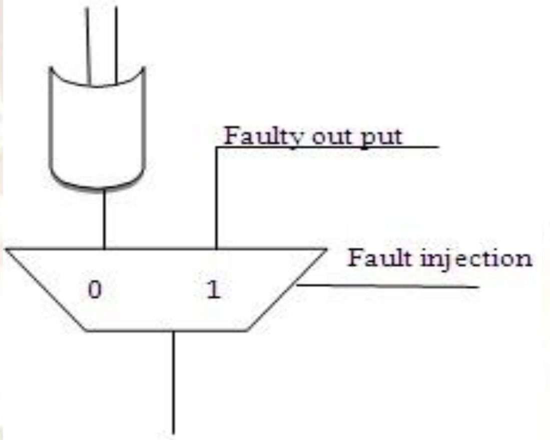

Fig.2: Fault injection in to gate.

The fault detection [8,9], procedure is always involves e to monitor the changes occurs at multiplier data whenever there is an attack [10], will takes place. In the proposed scheme the detection unit mainly contains three major blocks as described here, the basic multiplier, the error interpreter and the error detector.

Basic multiplier: Primary building block as described in section II.

Error Interpreter: This produces modulus operations on the two multiplier inputs.

Error Detector: Includes modulo operation on two multiplier inputs, selective element and a comparator unit.

The final arrangement is given in the following figure. The Booth encoded Wallace tree will produces the 40-bit product which is an input to the Error Detector, the Error interpreter whose inputs are modulus of primary inputs and with the help of a selective network any of modulo product can be given to the Detector, Error Detector unit contains again a modulo generator and a comparator network.

Whenever the fault is injected the predictor whose modulo operations are compared with another modulo values from original basic multiplier across the Error Detector, if the comparator output is zero which gives there is no fault injection else a fault injection can be noticed.

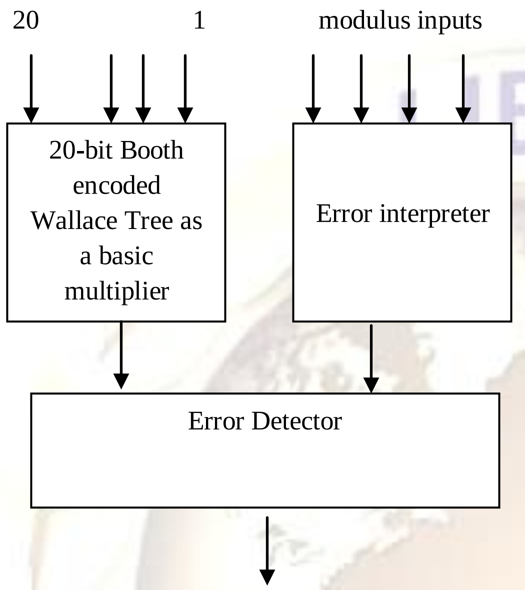

Fig.3 Fault Detection Model

As a consequence whenever there is a fault injection, correspondingly the product data may changes randomly, and depending on the bit size of the multiplier the error can be tracked with successive comparison technique. The description of this unit is done with Verilog and simulated and synthesized using Xilinx14.1; the results are analyzed in the next sections.

IV. SIMULATION AND SYNTHESIS

The Verilog RTL Description of the above article is simulated using Xilinx14.1 (ISE-Simulator), and results are observed separately with and without fault inputs, it is noticed that at some instance of simulation time the product values are differs with and without fault presence the various results are shown in the following figure.4 and figure.5.

When we analyze the simulation outputs it is observed that at similar instance of simulation time the data vales related to output product are different when fault is enabled.

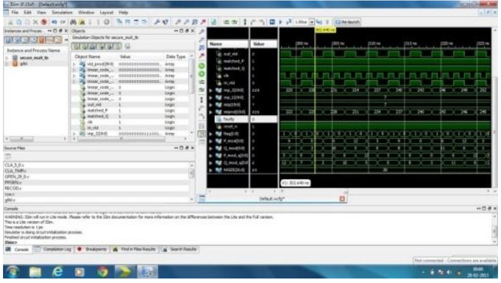

Fig.4. Simulation without Fault

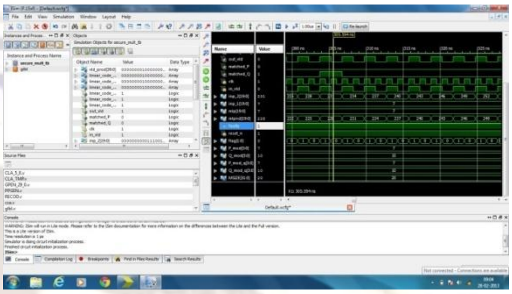

Fig.5 Simulation with Fault

The following synthesis report can help to analyze more when hardware units of Cryptographic devices are to be prepared the results mainly includes area, timing and delay reports as listed below. And the same are tested with advanced FPGA architecture the KINTEX7 evaluation platform. The synthesis is done with Xilinx14.1 and its reports are summarized below.

```
HDL Synthesis Report
=====================================
======
Macro Statistics
# Multipliers                                    :
2
 4x4-bit multiplier                              :
2
# Adders/Subtractions                            :
170
 4-bit adder                                     :
6
 5-bit adder                                     :
72
 6-bit adder                                     :
92


# Registers                                      :
7
  1-bit register                                 :
2
  20-bit register                                :
2
  40-bit register                                :
1
```

```
 8-bit register                                   :
2
# Comparators                                     :
3
 4-bit comparator equal                           :
3
# Multiplexers                                    :
486
 1-bit 2-to-1 multiplexer                         :
466
 21-bit 2-to-1 multiplexer                        :
10
 4-bit 2-to-1 multiplexer                         :
10
# XORs                                            :
498
 1-bit xor2                                       :
498
=====================================
======
Advanced HDL Synthesis Report
=====================================
======
Macro Statistics
#Multipliers                                      :
2
4x4 registered multiplier                         :
2
#Adders/Subtractors                               :
140
 4-bit adder                                      :
6
 5-bit adder                                      :
12
 5-bit adder-carry in                             :
30
 6-bit adder                                      :
92
#Registers                                        :
82
 Flip-Flops
: 82
#Comparators
: 3
 4-bit comparator Equal
: 3
#Multiplexers                               : 486
 1-bit 2-to-1 multiplexer                   : 466
 21-bit 2-to-1 multiplexer                  : 10
 4-bit 2-to-1 multiplexer                   : 10
#XORs                                       : 498
 1-bit xor2                                 : 498
=====================================
======
Design Summary
=====================================
======
Top Level Output File Name: secure_mult_top.ngc
Primitive and Black Box Usage:
```

```
--------------------------------------------------------------
---------
#     BELS                              : 1518
#     GND                               : 1
#     INV                               : 1
#     LUT2                              : 21
#     LUT3                              : 163
#     LUT4                              : 177
#     LUT5                              : 511
#     LUT6                              : 586
#     MUXCY                             : 12
#     MUXF7                             : 32
#     XORCY                             : 14
#     Flip Flops/Latches                : 124
#     FDC                               : 1
#     FDR                               : 57
#     FDRE                              : 66
#     Clock Buffers                     : 1
#     BUFGP                             : 1
#     IO Buffers                        : 227
#     IBUF                              : 43
#     OBUF                              : 184

Device utilization summary:
--------------------------------------------------------------
----------
Selected Device                         :
7k325tffg900-2
Slice Logic Utilization:
Number of Slice Registers               : 124/407600
Number of Slice LUTs                    :
1459/203800
Number used as Logic                    :
1459/203800
Slice Logic Distribution:
Number of LUT Flip Flop pairs used        : 1511
Number with an unused Flip Flop                    :
1387/1511
Number with an unused LUT                 :      52
/1511
Number of fully used LUT-FF pairs         :      72
/1511
Number of unique control sets             : 5
IO Utilization:
Number of IOs                             : 228
Number of bonded IOBs                     :
228/500
Specific Feature Utilization:
Number of BUFG/BUFGCTRLs                  : 1/32


--------------------------------------------------------------
----------
Partition Resource Summary:
--------------------------------------------------------------
----------
No Partitions were found in this design.
```

```
======================================
======
Timing Report
======================================
======
NOTE: THESE TIMING NUMBERS ARE ONLY
A SYNTHESIS ESTIMATE.FOR ACCURATE
TIMING INFORMATION PLEASE REFER TO
THE TRACE REPORT GENERATED AFTER
PLACE-and-ROUTE.

Clock Information:
----------------------------------------------------------------
----------
Clock Signal Clock buffer (FF name) | Load
Clk BUFGP              | 124  |
----------------------------------------------------------------
----------
Asynchronous Control Signals Information:
----------------------------------------------------------------
----------
No asynchronous control signals found in this design

Timing Summary:
----------------------------------------------------------------
----------
Speed Grade                                      : -2
Minimum period                                   :
10.792ns
Maximum Frequency                                :
92.658MHz
Minimum input arrival time before clock          :
9.110ns
Maximum output required time after clock         :
39.763ns
Maximum combinational path delay                 : No
path found
Timing Details:
----------------------------------------------------------------
----------
All values displayed in nanoseconds (ns)
======================================
======
Timing constraint            : Default period analysis
for Clock 'clk'
Clock period                 : 10.792ns
Frequency                    : 92.658MHz
Total number of paths / destination ports:
55969378809 / 59
Delay                        : 10.792ns
(Levels of Logic = 21)
Source                       : a_in_16 (FF)
Destination                  : U_mul2/Mmult_ab_0
(FF)
Source Clock                 : clk rising
Destination Clock            : clk rising

Cross Clock Domains Report:
----------------------------------------------------------------
----------
Clock to Setup on destination clock clk
|Src:Rise|Src:Fall|Src:Rise|Src:Fall|Source Clock
|Dest:Rise|Dest:Rise|Dest:Fall|Dest:Fall|
----------------------------------------------------------------
----------
Clk | 10.792|   |     |     |
----------------------------------------------------------------
----------
Total REAL time to XST completion    : 49.00 Sec
Total CPU time to XST completion     : 48.95 Sec
Total memory usage                   :    497284
                                kilobytes
Number of errors                     :    0    (0
filtered)
Number of warnings                   :    0    (0
filtered)
Number of infos                      :       15(0
filtered)
```

The corresponding schematics are shown in the following figures.

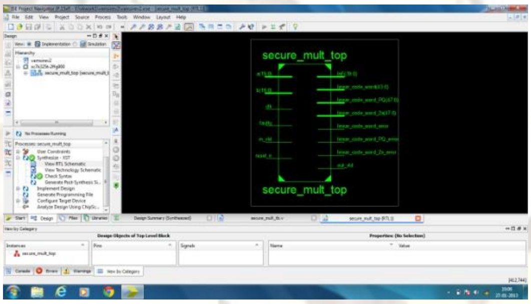

Fig.6 RTL Schematic Diagram

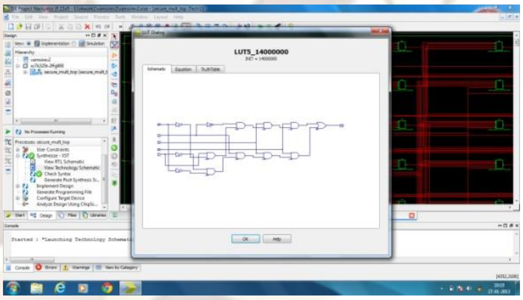

Fig.7 Technology Schematic Diagram

V. FPGA SETUP

The final section shows the results observed on KINTEX7-FPGA evaluation board the relevant configuration, its connection with working platform and the output observation with Chip Scope Pro is shown in the following figures.

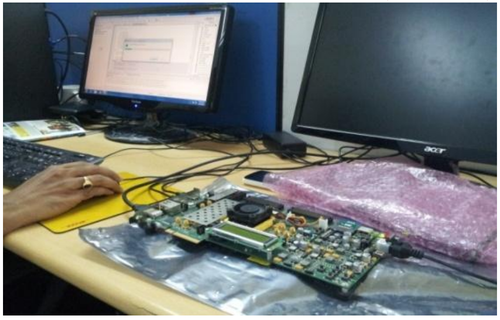

Fig.8 KINTEX7 FPGA Module Setup

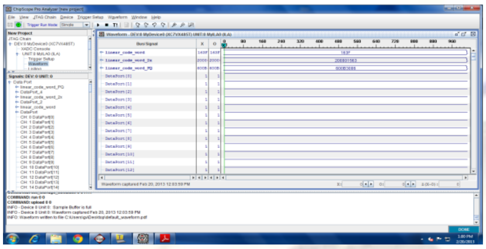

Fig.9 FPGA Output wave form on Chip Scope Pro

VI. CONCLUSION

This paper analyzes the simulation and FPGA modeling of Fault injection attacks on Cryptographic Devices, the modified Booth encoded Wallace tree multiplier based fault detection unit is more reliable and good sensitive to Fault injection attacks, the multi modulus multi liner arithmetic block are superior and good counter parts compared to existed proposals. The hardware realization with these techniques are best measures to design more advanced and secured architectures to notice quickly the intruder attacks on the Cryptographic Devices and is possible to take necessary actions to prevent the malfunctions on Commercial Electronics.

ACKNOWLEDGMENT

We would like to thank Dr.Fazal NoorBasha, for his out standing support and also we would like to especially express gratitude "EDUPLUS" team for their technical advices.

REFERENCES

[1]. Zhen Wang, Mark Karpovsky and Ajay Joshi "*Secure Multipliers Resilient to Strong Fault-Injection attacks using Multilinear Arithmetic Codes", IEEE* – 2011.

[2]. G.Canivet, P. Maistri, R. Leveugle, J.Cldire, F. Valette, and M. Renaudin, "*Glitch and laser fault attacks on to a secure AES implementation on a SRAM – based FPGA*", J.Cryptol., vol.24, no.2, PP. 1-22, Apr.2011

[3]. E.Trichina and R. Korkikyan, "*Multi fault laser attacks on protected CRT – RSA*", in proc.workshop on fault diagnosis tolerance Cryptography, 2010, pp.75-86

[4]. K.D.Akdemir, Z.Wang, M.G.Karpovsky and B.Sunar, "*Design o f Cryptographic devices resilient to fault injection attacks using nonlinear robust codes*", in fault analysis in Cryptography. Newyork: Springer- verlag, 2011.

[5]. C.Wallace "*A suggestion for a fast multiplier*," IEEE Trans.Electron.Comput, vol.EC-13, no, 1, pp.14-17, feb.1964.

[6]. K.J.Kulikoski,M.G.Karpovsky, and A.Taubin "*Robust codes and Robust,fault tolerant architectures of the advanced encryption standard*" .Journal of systems Architecture,53:138-159,2007.

[7]. Z.Wang, M.G.Karpovsky, B.Sunar and A.Joshi, "*Design of Reliable and secure multipliers by Multilinear arithmetic codes*", information and communication security, ser.lec.notes in computer science, vol. 5927, pp.47-62, 2009

[8]. A.Krasniewski, "*Concurrent error detection for finite state machines implemented with embedded memory blocks of SRAM-based FPGA*'s," Microprocessors and Microsystems, 2008.

[9]. C.H.Kim and J.J. Quisquater, "*How can we overcome bothside channel analysis and fault attacks on RSA-CRT?"* in FDTC'07: proceedings of the workshop on fault diagnosis and tolerance in Cryptography,Washington,DC,USA :IEEE computer society,2007,pp.21-29.

[10]. J.M. Schmidt and M.Hutter, "*Optical and EM fault attacks on CRT-based RSA:Concrete results*,"in proc.15th Austrian Workshop Mi-croelectron..,2007,pp.75-86.

[11]. B.Skoric,S.Maubach,T.Kevenaar,and, P.Tuyls. "*Information-theoretic Analysis of Coating PUF's.Cryptology*", eprint Archive,report 2006/101,2006.

[12]. H.bar- l,H.Choukri,D.Naccache,M.Tunstall, and C.Whelan, "*The Sorcerer's apprentice guide to fault attacks,*" proc.IEEE,vol.94,no.2,pp.370-382,feb 2006.

[13]. I.Vasyltsov,E.Hambardzumyan,y.-s.Kim,and B.Karpinskyy,"*Fast digital TRNG based on metastable ring Oscillator,*"in.proc.Cryptograph.hardw.Embed.syst.workshop(CHES),2008,pp.164-180.

[14]. Nangate lnc.,Sunnyvale,CA, "*Nangate 45nm opencell library,*" 2009.[0nline].Available:http://www.nangate.com.

[15]. D.Roberts,T.Austin,D.Blauww,T.Mudge,and K.Flautner, "*Error analysis fro the support of robust voltage scaling,*" in proc.6th lnt.Symp.Quality Electron .design(ISQED),2005,pp.65-70.