American Journal of Applied Sciences 11 (1): 137-144, 2014
ISSN: 1546-9239

doi:10.3844/ajassp.2014.137.144 Published Online 11 (1) 2014 (http://www.thescipub.com/ajas.toc)

ERROR DETECTION SCHEMES FOR FINITE FIELD MULTIPLIERS

[1]Sargunam, B. and [2]R. Dhanasekaran

[1]Department of Electronics and Communication Engineering, Avinashilingam University, Coimbatore, India
[2]Syed Ammal Engineering College, Ramanathapuram, India

Received 2013-09-16, Revised 2013-10-24; Accepted 2013-12-11

ABSTRACT

Finite field multipliers are widely used in the field of cryptography for the purpose of scalar multiplication. The outputs of the finite field multipliers may consist of errors due to certain natural radiations which further leads to the failure of the cryptosystems. Here two Concurrent Error Detection (CED) schemes namely time redundancy and modular inversion based error detection schemes for finite field multipliers are discussed. The CED techniques have been implemented for bit serial, digit serial and bit parallel Montgomery multipliers. The Simulation results are obtained using Modelsim10.0b, area and power analysis has been performed using Xilinx ISE 9.1i. The proposed modular inversion based CED scheme is found to be area and power efficient compared to existing time redundancy based CED scheme.

Keywords: Montgomery Multiplication, Elliptic Curve Cryptography (ECC), Parity Prediction, Modular Inversion, Finite Field Multipliers

1. INTRODUCTION

The finite field multiplication has received great attention in literature (Lee *et al*., 2006; Ghosh *et al*., 2011) among the basic operations. It is mainly because the implementation of a multiplier is much more complex when compared to adder and by using multiplication operation repeatedly one can perform difficult field operations such as inversion and exponentiation which are widely used in cryptosystems. Finite field popularly known as Galois Field (GF) is represented as GF (p^n), where p^n is a prime number over 'n' dimensions. When the prime number is 2, elements of GF are expressed as binary numbers. GF (2) when extended to GF (2^m) is termed as binary extension field. Since no carry propagation occurs in GF (2^m), the addition of two single bits requires only a logical XOR operation.

Finite fields are used in a variety of applications including classical coding theory in linear block codes such as Reed Solomon codes and in cryptographic algorithms (MacWilliams and Sloane, 1998). Cryptography is the practice and study of techniques for secured data communication in the presence of third parties. ECC (Miller, 1998) is an approach to public key cryptography based on algebraic structure of elliptic curves over finite field. This cryptographic method has been regarded mature to provide robustness for secure data transaction. Therefore ECC has become an attractive alternative cryptosystem and many designs have been proposed in recent years (Sakiyama *et al*., 2007; Chumg *et al*., 2005; Gura *et al*., 2002; Blake *et al*., 2005; Biham and Shamir, 1997; Boneh *et al*., 1997).The Montgomery multiplication algorithm is used to enhance the scalar multiplication in ECC (Montgomery, 1985).

CED is a process used to detect the errors in a cryptosystem while the system is performing its data transmission operation (Mitra and McCluskey, 2000; Reyhani-Masoleh and Hasan, 2006; Hariri and Reyhani-Masoleh, 2007; Bayat-Sarmadi and Hasan, 2007). Due to the fact that fault injection and active attacks are used against cryptosystems, it is very important to increase the reliability of the elliptic curve-based cryptosystems and in particular, its main arithmetic operation, i.e., multiplication. The presence of fault in cryptosystems

Corresponding Author: Sargunam, B., Department of Electronics and Communication Engineering, Avinashilingam University, Coimbatore, India

can lead to an active attack which results in leakage of secret information from the cryptosystems. The simplest way to prevent such an attack is to ensure that the computational device, the multiplier, verifies the value it computes before sending them out. To meet this purpose concurrent error detection scheme could be one of the options to mitigate logic errors. The design of efficient multipliers with CED capability is desirable to have a highly reliable and dedicated cryptographic hardware (Hariri and Reyhani-Masoleh, 2011).

Finite field multipliers use Montgomery multiplication algorithm to perform bit serial, digit serial and bit parallel multiplier operations (Ananyi *et al*., 2009; Koc and Acar, 1998; Fan and Dai, 2005; Hariri and Reyhani-Masoleh, 2008). The finite field elements are represented using three basis representations namely polynomial basis, normal basis and dual basis. Polynomial basis has found to be suitable for the purpose of error detection as conversion from polynomial basis to binary is quite simple. The bit parallel systolic finite field multiplier over polynomial basis has been implemented for irreducible polynomial, all-one polynomial and irreducible trinomial (Sargunam *et al*., 2012a). The speed of bit parallel systolic finite field multiplier over polynomial basis has been improved using an unique technique (Sargunam *et al*., 2012b). Reyhani-Masoleh and Hasan (2003) a parity prediction based technique has been implemented for a polynomial basis multiplier. The major drawback of this technique was that the exact error bit position was not specified in the output of the multiplier instead only the existence of error was detected. In this study two error detection schemes have been discussed, the time redundancy and the modular inversion based error detection techniques.

2. TIME REDUNDANCY TECHNIQUE

The fault attacks are common against cryptographic algorithms. CED is one of the counter measures used to protect the crypto-processors in case of such attacks. In this section, we discuss CED circuits for bit-serial, digit-serial and bit-parallel Montgomery multipliers which can be used as a counter measure against natural faults and fault attacks in cryptography.

2.1 Time Redundancy Approach

The architecture using time redundancy can avoid the potential security problem caused by side-channel attacks. All single cell faults in the multiplier will be concurrently detected. Moreover, this multiplier requires a little space overhead and takes only few extra clock cycles. This technique is applied for bit serial, digit serial and bit parallel multipliers. The block diagram for the time redundancy approach is shown in the **Fig. 1**. The latches are used to store the data and 2-to-1 Mux is a 2 by 1 multiplexer to select one of the inputs.

CED using time redundancy technique is as follows:

$A.x^m$ = mod F(x) | $B.x^m$ mod F(x)
C = A.B mod F(x)
C' = A'.B' x^{-m}mod F(x)
C' = A'.B' x^{-m} mod F(x)
= (A. x^m).(B.x^m) x^{-m} mod F(x)
= A.B x^m mod F(x)
= **C** x^m mod F(x)

The fundamental operation of the multiplier is explained in the following steps.

The first step is performed by applying inputs A(x) and B(x) to the Montgomery Multiplier array and the result C(x) is converted by the *x^m circuit to C'(x) and stored in latches. The dataflow of this first step is shown in bold lines in **Fig. 2**.

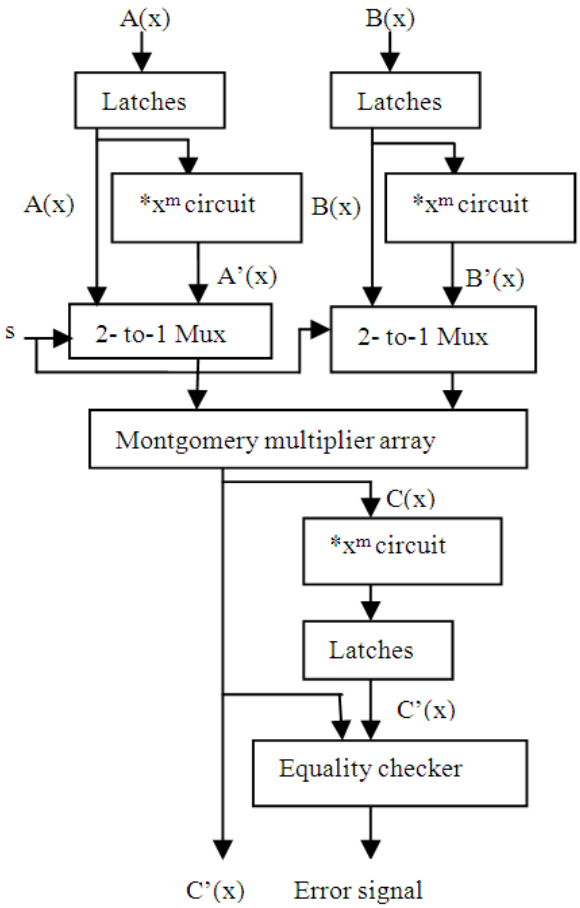

Fig. 1. CED using time redundancy (Chiou *et al*., 2006)

The second step is executed by applying inputs A'(x) and B'(x) to the Montgomery Multiplier array. The inputs A(x) and B(x) are applied to respective $*x^m$ circuits to obtain A'(x) and B'(x). The result C'(x) is compared to the previously stored result C'(x) in latches.

The function unit $*x^m$ realizes the following function $Q'(x) = Q(x)*x^m \bmod P(x)$. Where Q(x) and Q'(x) are the inputs and output of the $*x^m$ circuit respectively. There is one to one correspondence between Q(x) and its Q'(x) in residue representation. The dataflow of the second step is shown in bold lines in **Fig. 3**. The C'(x) values obtained from step 1 and step 2 are compared using equality checker and the error signal is produced. The outputs of both these steps are equal no error signal is generated and if not the error signal is generated to indicate the error.

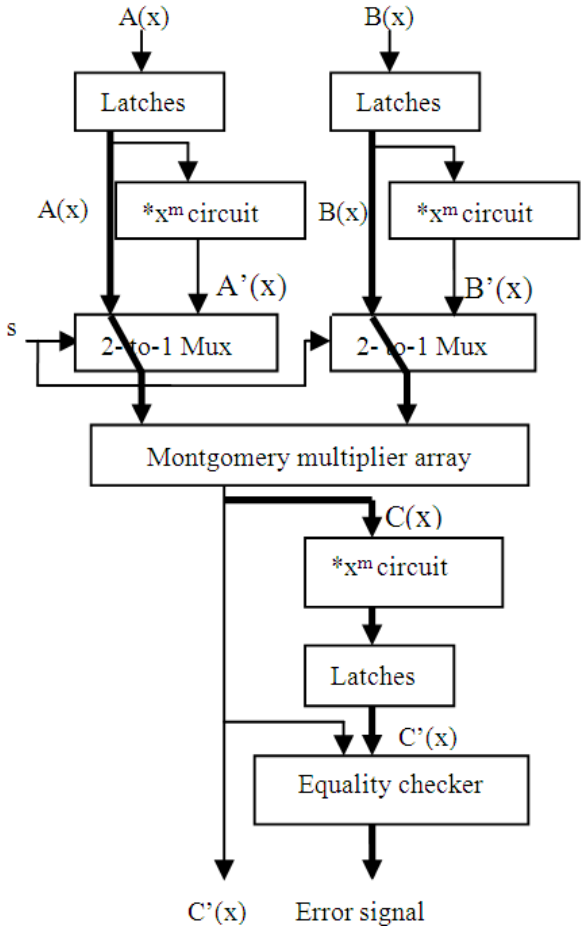

Fig. 2. The data flow in the time redundancy technique during the 1st Step

By examining the error signal at the output of equality checker the errors are detected. The exact error bit position is also detected by this method.

3. MODULAR INVERSION TECHNIQUE

It was found that the parity prediction technique failed to detect the exact bit positions of the erroneous output of the multipliers and this technique was not efficient to detect the online errors that occurred in the cryptosystems. In (13) a time redundancy scheme was developed for the purpose of CED using modular multiplication. There are two important performance criteria in VLSI implementation, namely power and area. Trade-off may exist between the two parameters. Optimization of these two parameters can be carried out in finite field multiplier architecture in order to consume low power and low area. The time redundancy scheme was found to have high power and area utilization. In order to attain a power and area efficient CED scheme modular inversion algorithm has been used.

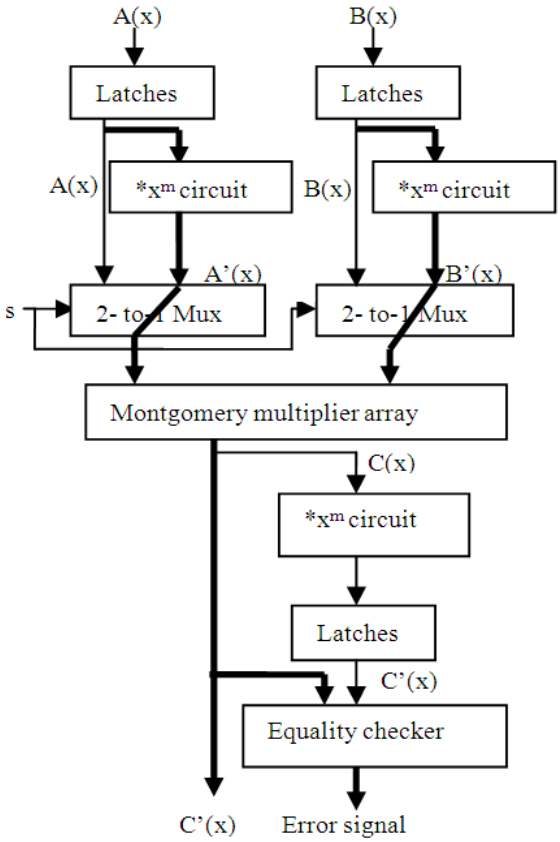

Fig. 3. The data flow in time redundancy technique during the 2nd step

3.1 Modular Inversion

The multiplication inversion of an element aεF is defined as the process to find an element a^{-1}εF, such that $a.a^{-1} = 1 \bmod P(x)$. Several algorithms to compute the multiplicative inverse in GF (2^m) have been proposed in literature. The inverse is computed using an improved modification of the extended Euclidian algorithm called modular inversion algorithm. The modular multiplicative inverse a^{-1} (mod p) of an integer 'a' exists if and only if 'a' and 'p' are relatively prime, that is gcd (a,p) = 1. In all cases considered, p is prime and hence 'a' and 'p' are always relatively prime. The following is the modular inversion algorithm that has been incorporated in the CED scheme.

Algorithm

Inputs: Operand a, prime p
Output: a^{-1} mod p
Step1: u = a, v = p, $x_1 = 1$, $x_2 = 0$
Step2: while $u \neq 1$ and $v \neq 1$ do
Step 2.1: while u even do
Step2.1.1: u = u/2
Step 2.1.2: if x_1 even then $x_1 = x_1/2$
else $x_1 = (x_1 + p)/2$
Step 2.2: while v even do
Step 2.2.1: v = v/2
Step 2.2.2: if x_2 even then $x_2 = x_2/2$
else $x_2 = (x_2 + p)/2$
Step 2.3: if $u \geq v$ then $u = u - v$, $x_1 = x_1 - x_2$
else $v = v - u$, $x_2 = x_2 - x_1$
Step 3: if u = 1 then return x_1(mod p)
else return x_2 (mod p)

The step 2 of the algorithm runs iteratively and proceeds towards the goal. In this step for every iteration either 'u' or 'v' is reduced by at least one bit length. The total number of iterations in step 2 is at most 2k, where k is the maximum bit length of 'p' and 'a'.

3.2 Error Detection Method

In order to obtain an efficient CED scheme for the purpose of detecting errors in the output of the finite field multipliers the modular inversion algorithm has been incorporated into the error detecting scheme. This technique has been proved to have better power and area efficiency when compared to the time redundancy scheme. The block diagram for modular inversion technique is shown in **Fig. 4**. The modular inversion technique also performed in two steps.

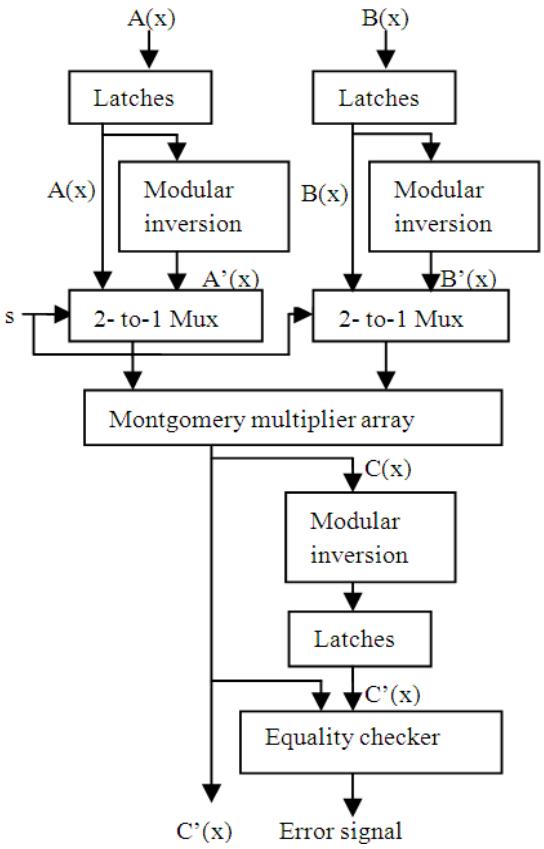

Fig. 4. Modular inversion based error detection scheme

The multiplication array block performs bit serial, digit serial or bit parallel multiplication in finite field. The 2-to-1 Mux block selects one of the inputs for multiplication based on the select signal 'S'. The error detection process is performed using the block diagram by multiplying two inputs A(x) and B(x). Instead of modular multiplication in time redundancy technique here modular inversion is used to detect the errors. In this technique also exact error bit position can be detected and it can detect multiple errors.

The data flow for the CED scheme using modular inversion in the block diagram is explained in two steps as follows:

During the first step the two inputs (A(x), B(x)) are multiplied using the Montgomery multiplication algorithms (Bit serial, Digit serial or Bit Parallel). The output of the Montgomery multiplication array (C(x)) is further taken as input into the modular inversion block where the inversion algorithm is performed and the output C'(x) is generated. The blocks which are used and the data flow during this first step is shown in **Fig. 5**.

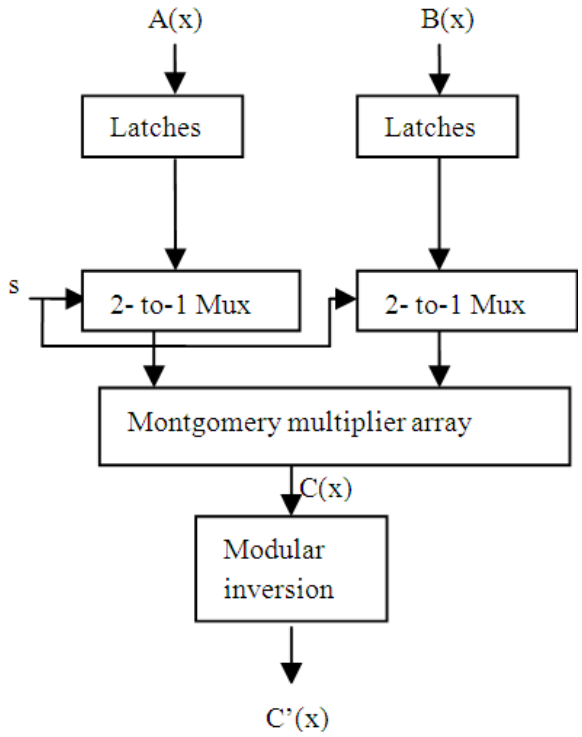

Fig. 5. The data flow in the modular inversion technique during the 1st Step

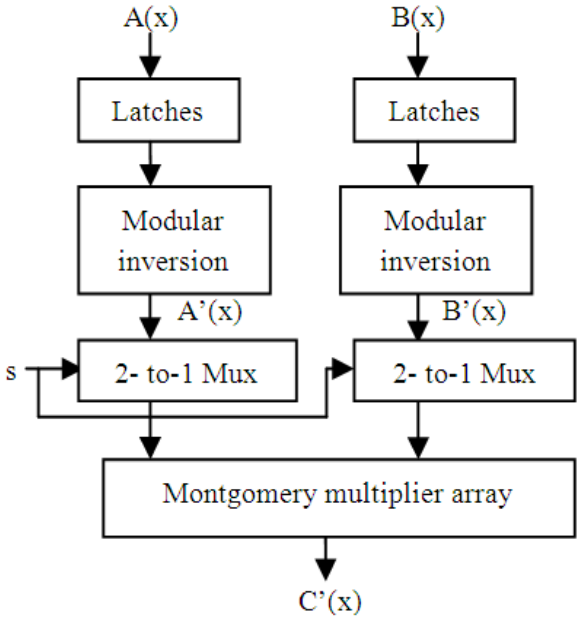

Fig. 6. The data flow in modular inversion technique during the 2nd step

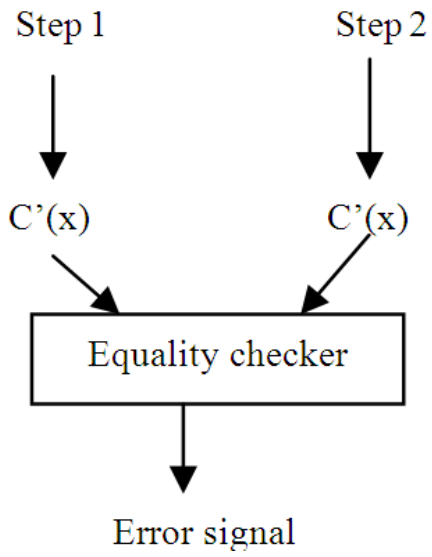

Fig. 7. Comparison of the outputs from step 1 and step 2 using equality checker

During the second step the two inputs A(x) and B(x) are individually inverted using the modular inversion algorithm to form A'(x) and B'(x). The inverted outputs are taken into the Montgomery multiplication array and multiplied using the Montgomery multiplication algorithms (Bit serial, Digit serial or Bit Parallel). The output from the Montgomery multiplication array is generated as C'(x). The blocks which are used for this step and the data flow are shown in the **Fig. 6**.

The outputs of step 1 and 2 (C'(x)) are compared in the equality checker. If the outputs of the two steps are different the error signal is generated as shown in **Fig. 7**. The existence of error and the error bit positions can be identified by examining the output of the equality checker.

4. IMPLEMENTATION RESULTS

The algorithms for the time redundancy and the modular inversion error detection technique have been coded using VHDL and simulated using Mentor Graphics front end (Modelsim 10.0b). The implementation is done using Xilinx ISE 9.1i and area and power reports are obtained. The bit serial, digit serial and bit parallel Montgomery multipliers are coded and the time redundancy and modular inversion techniques are applied for all the multiplier types.

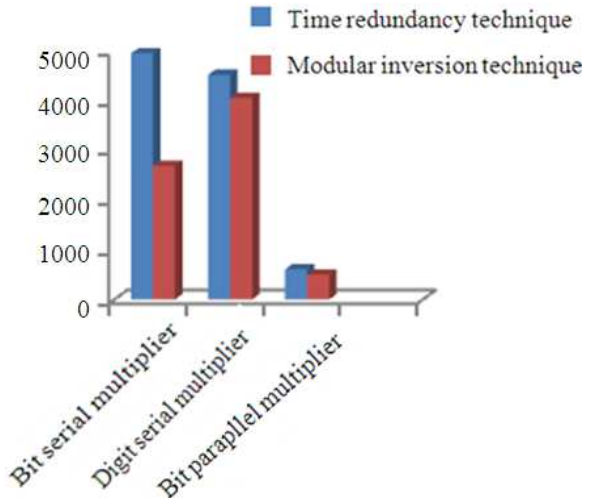

Fig. 8. Comparison of time redundancy technique and modular inversion technique in terms of gate count

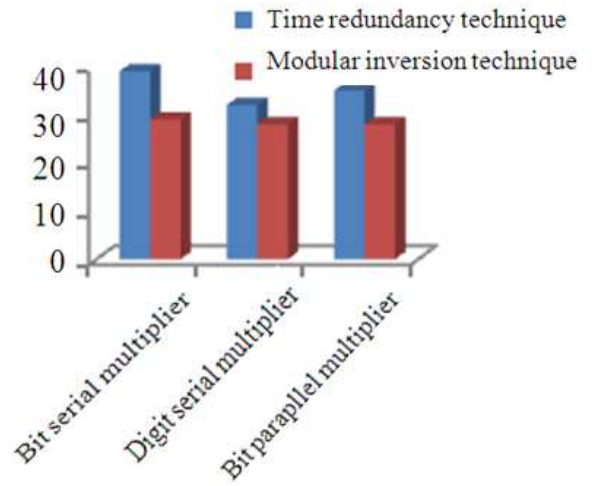

Fig. 9. Comparison of time redundancy technique and the modular inversion technique in terms of power consumption in Mw

Figure 8 and 9 show the graphical comparison of the area and power consumption of the time redundancy technique and the modular inversion based error detection technique for all the three multiplier types. **Figure 10** shows the simulation result for the error detection in bit serial multiplier using time redundancy technique.

Figure 11 shows the simulation result for the error detection in bit serial multiplier using the modular inversion based error detection scheme.

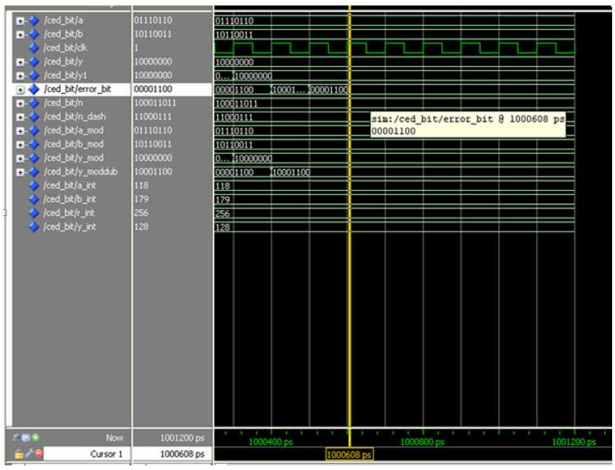

Fig. 10. Simulation result for the error detection in bit serial multiplier using time redundancy technique

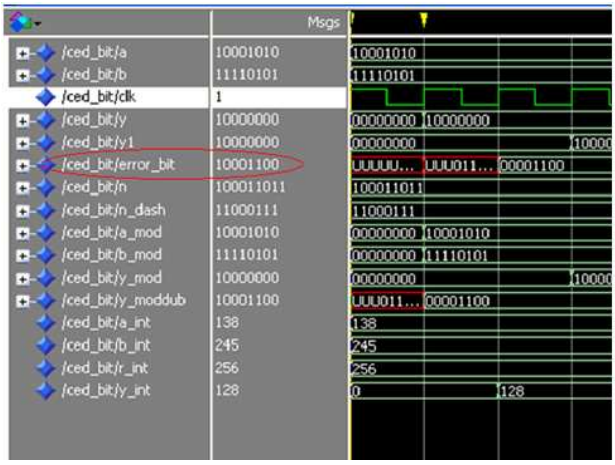

Fig. 11. Simulation result for the error detection in bit serial multiplier using the modular inversion scheme

5. CONCLUSION

The CED scheme is used to detect online errors in applications like cryptography. The time redundancy and modular inversion based CED schemes are performed for the three types (Bit-serial, Digit-serial and Bit-parallel) of finite field multipliers using Montgomery multiplication algorithm. The proposed CED using modular inversion technique is found to be area and power efficient when compared to the time redundancy technique.

6. REFERENCES

Bayat-Sarmadi, S. and M. Hasan, 2007. On concurrent detection of errors in polynomial basis multiplication. IEEE Trans. Very Large Scale Integ. Syst., 15: 413-426. DOI: 10.1109/TVLSI.2007.893659

Biham, E. and A. Shamir, 1997. Differential fault analysis of secret key crypto systems. Proceedings of the 17th Annual International Cryptology Conference Santa Barbara, Aug. 17-21, California, USA, pp. 513-525. DOI: 10.1007/BFb0052259

Blake, I., G. Seroussi and N. Smart, 2005. Advances in Elliptic Curve Cryptography. New York: Cambridge,

Boneh, D., R.A. Demillo and R.J. Lipton, 1997. On the importance of checking cryptographic protocols for faults. Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques Konstanz, May, 11-15, Germany, pp: 37-51. DOI: 10.1007/3-540-69053-0_4

Chiou, C.W., C.Y. Lee, A. Wen Deng and J.M. Lin, 2006. Concurrent error detection in montgomery multiplication over GF (2^m). IEICE Trans. Fundamentals Electron. Commun. Comput. Sci., 2: 566-574. DOI: 10.1093/ietfec/e89-a.2.566

Chumg, R.C.C., N.J. Telle, W. Luk and P.Y.K. Cheung, 2005. Customizable elliptic curve cryptosystems. IEEE Trans. Very Large Scale Integr., 13: 1048-1058. DOI: 10.1109/TVLSI.2005.857179

Fan, H. and Y. Dai, 2005. Fast bit-parallel GF(2^n) multiplier for all trinomials. IEEE Trans. Comput., 54: 485-490. DOI: 10.1109/TC.2005.64

Gura, N., S.C. Shantz, H. Eberle, S. Gupta and V. Gupta *et al.*, 2002. An end-to-end systems approach to elliptic curve cryptography. Proceedings of the 4th International Workshop on Cryptographic Hardware and Embedded Systems, Aug. 13-15, London, UK., 349-365. DOI: 10.1007/3-540-36400-5_26

Hariri, A. and A. Reyhani-Masoleh, 2007. Fault detection structures for the Montgomery multiplication over binary extension fields. Proceedings of the Workshop Fault Diagnosis and Tolerance in Cryptography, Sep. 10-10, IEEE Xplore Press, Vienna, pp: 37-46. DOI: 10.1109/FDTC.2007.19

Hariri, A. and A. Reyhani-Masoleh, 2008. Digit serial structures for the shifted polynomial basis multiplication over binary extension fields. 2nd International Workshop, Jul. 6-9, Siena, Italy, pp: 103-116. DOI: 10.1007/978-3-540-69499-1_9

Hariri, A. and A. Reyhani-Masoleh, 2011. Concurrent error detection in montgomery multiplicaton over binary extension fields. IEEE Trans. Comput., 60: 1341-1353. DOI: 10.1109/TC.2010.258

Ananyi, K., H. Alrimeih and D. Rakhmatov, 2009. Flexible hardware processor for elliptic curve cryptography over nist prime fields. IEEE Trans. Very Large Scale Integ. Syst., 17: 1099-1112. DOI: 10.1109/TVLSI.2009.2019415

Koc, C.K. and T. Acar, 1998. Montgomery multiplication in GF (2^k). Designs Codes Cryptography, 14: 57-69. DOI: 10.1023/A:1008208521515

Lee, C.Y., C.W. Chiou and J.M. Lin, 2006. Concurrent error detection in a polynomial basis multiplier over GF(2^m). J. Electron. Test. Theory Appli., 22: 143-150. DOI: 10.1007/s10836-006-7446-9

MacWilliams, F.J. and N.J.A. Sloane, 1998. The Theory of Error-Correcting Codes. 1st Edn., North-Holland New York, ISBN-10: 0444851933, pp: 762.

Miller, V.S., 1998. Use of elliptic curves in cryptography. Adv. Cryptol. Crypto, 85: 417-426. DOI: 10.1007/3-540-39799-X_31

Mitra, S. and E.J. McCluskey, 2000. Which concurrent error detection scheme to choose? Proceedings of the International Test Conference, Oct. 3-5, IEEE Xplore Press, Atlantic City, NJ, pp: 985-994. DOI: 10.1109/TEST.2000.894311

Montgomery, P.L., 1985. Modular multiplication without trial division. Math. Comput., 44: 519-521.

Ghosh, S., D. Mukhopadhyay and D. Roychowdhury, 2011. Petrel: Power and timing attack resistant elliptic curve scalar multiplier based on programmable GF(p) arithmetic unit. IEEE Trans. Circuits Syst., 58: 1798-1812. DOI: 10.1109/TCSI.2010.2103190

Reyhani-Masoleh, A.R. and M. Hasan, 2003. Error detection in polynomial basis multipliers over Binary extension fields. Proceedings of the 4th International Workshop Redwood Shores, Aug. 13-15, IEEE Xplore Press, CA, USA, pp: 515-528. DOI: 10.1007/3-540-36400-5_37

Reyhani-Masoleh, A. and M.A. Hasan, 2006. Fault detection architectures for field multiplication using polynomial bases. IEEE Trans. Comput., 55:1089-1103. DOI: 10.1109/TC.2006.147

Sakiyama, K., L.B atina, B.Preneel and I. Verbauwhede, 2007. High-performance Public-key cryptoprocessor for wireless mobile applications. Mobile Netw. Appli., 12: 245-258. DOI: 10.1007/s11036-007-0020-6

Sargunam, B., S. Arul Mozhi and R. Dhanasekaran, 2012a. Efficient bit-parallel systolic multiplier for special class of GF(2^m). Proceedings of the International Conference on Electrical, Electronics Computer Engineering, (ECE' 12), IEEE Xplore Press, Ahmedabad.

Sargunam, B., S. Arul Mozhi and R. Dhanasekaran, 2012b. High speed bit-parallel systolic multiplier over GF(2^m) for cryptographic application. IEEE Proceedings of the International Conference Advanced Communication Control Computing Technologies, Aug. 23-25, IEEE Xplore Press, Ramanathapuram, pp: 244-247. DOI: 10.1109/ICACCCT.2012.6320779