Concurrent error detection in bit-serial normal basis multiplication over $GF(2^m)$ using multiple parity prediction schemes

Lee, Chiou Yng; Meher, Pramod Kumar; Patra, Jagdish Chandra

2010

Lee, C. Y., Meher, P. K., & Patra, J. C. (2010). Concurrent Error Detection in Bit-Serial Normal Basis Multiplication Over $GF(2^m)$ Using Multiple Parity Prediction Schemes. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 18(8), 1234-1238.

<https://hdl.handle.net/10356/94346>

<https://doi.org/10.1109/TVLSI.2009.2020593>

© 2009 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. The published version is available at: [DOI: <http://dx.doi.org/10.1109/TVLSI.2009.2020593>].

Downloaded on 26 Apr 2023 19:47:45 SGT

Concurrent Error Detection in Bit-Serial Normal Basis Multiplication Over GF(2^m) Using Multiple Parity Prediction Schemes

Chiou-Yng Lee, Pramod Kumar Meher, and Jagdish Chandra Patra

C.-Y. Lee is with the Department of Computer Information and Network Engineering,
Lunghwa University of Science and Technology, Taoyuan 33306, Taiwan
(e-mail: pp010@mail.lhu.edu.tw).

P. K. Meher is with the Department of Communication Systems,
Institute for Infocomm Research, 138632 Singapore (e-mail: pkmeher@i2r.a-star.edu.sg).

J. C. Patra is with the School of Computer Engineering, Nanyang Technological University,
639798 Singapore (e-mail: aspatra@ntu.edu.sg).

Abstract

New bit-serial architectures with concurrent error detection capability are presented to detect erroneous outputs in bit-serial normal basis multipliers over GF(2^m) using single and multiple-parity prediction schemes. It is shown that different types of normal basis multipliers could be realized by similar architectures. The proposed architectures can detect errors with nearly 100% probability.

Index Terms—Concurrent error detection (CED), cryptography, normal basis, parity prediction.

I. INTRODUCTION

Among the finite field arithmetic operations over GF(2^m), multiplication is an important operation, which is complex, and time consuming. The hardware implementation of a multiplier of large orders requires large number of transistors, and it is likely that some of those transistors may go wrong which would lead to possible faults in the computation. Since, the reliability of computation is an important issue, research works have been addressed to concurrent error detection (CED) in digital circuits. It is generally accepted that the parity prediction technique provides an economical choice for detecting both temporary and permanent faults [1], [2]. The previous works on fault detection architecture are derived either by parity prediction method or by time redundant method. Time redundancy employs only a single set of hardware to carry out the operation repeatedly. Since the same hardware is used, the repetition of the operation is expected to produce the same erroneous results in the presence of faults [3], [4]. The parity prediction method on the other hand uses parity-check bit(s) to detect errors in the results. A family of code, such as Berger codes [13], has been developed for detecting t unidirectional errors and requires fewer parity check bits. For finite field GF(2^m), this approach is suitable for implementing CED multipliers with a single parity prediction scheme (as seen in [5] and [6]). CED multipliers with multiple parity prediction schemes are also found in [7]–[10]. In this paper, we propose a new bit-serial normal basis (NB) multiplier using parity prediction scheme with a simple and efficient architecture, which is suitable for all types of NB multipliers.

II. PRELIMINARIES

A. Bit-Serial Normal Basis Multiplication Over $GF(2^m)$

Assuming α to be a normal element of $GF(2^m)$, any element A in $GF(2^m)$ is given by $A = \sum_{i=0}^{m-1} a_i \alpha^{2^i} = (a_0, a_1, \dots, a_{m-1})$, where $a_i \in GF(2)$ for $0 \leq i \leq m-1$; and $N = \{\alpha, \alpha^2, \dots, \alpha^{2^{m-1}}\}$ is the normal basis of $GF(2^m)$. Let A and B be two NB elements in $GF(2^m)$. Using the like-polynomial basis multiplication scheme [11], the product C of A and B can be calculated as

$$C = \left(\left(\left(a_{m-1} \alpha B^{2^{-(m-1)}} \right)^2 + a_{m-2} \alpha B^{2^{-(m-2)}} \right)^2 + \dots \right)^2 + a_0 \alpha B \quad (1)$$

and can be computed recursively as

$$\begin{aligned} C_0 &= 0 \\ C_i &= C_{i-1}^2 + a_{m-i} \alpha B^{2^{-(m-i)}} \\ C &= C_m. \end{aligned} \quad (2)$$

As shown in Fig. 1 the bit-serial NB multiplier consists of two registers, one α module and a PS module. Registers K and Y are initialized by $B^{2^{-(m-i)}}$ and 0, respectively. During each cycle, the α module calculates αK ; and the PS module computes $C_i = C_{i-1}^2 + a_{m-i} \alpha B^{2^{-(m-i)}}$. After the m cycles, the product C appears at the output register Y .

B. Parity Prediction Scheme

In its basic form, the single-parity prediction scheme associates m data bits with a parity bit. The fault model of the NB multiplier is shown in Fig. 2. The CED circuit includes the multiplier circuit under test with a parity generator and an equality checker. The parity generator uses A and B to obtain the parity bit $\hat{P}_C$ of C . The equality checker computes $\hat{e}_C = \hat{P}_C + P_C$ to compare the parity of the result $P_C = \sum_{i=0}^{m-1} c_i \bmod 2$ with the predicted parity P_C . $\hat{e}_C = 1$ indicates stuck-at fault in the multiplier.

III. PROPOSED BIT-SERIAL NORMAL BASIS MULTIPLIER WITH CED CAPABILITY

We discuss here the parity prediction functions of α module and a PS module to derive the proposed bit-serial normal basis multiplier over $GF(2^m)$ with CED capability, based on a single fault model.

A. Parity Prediction of α Module

As shown in [12], $\alpha = \sum_{i=0}^{t-1} \gamma^{2^{im}}$ is a special case of normal basis element, called the type- t Gaussian normal basis (GNB), if $p = mt + 1$ be a prime number and $\gcd((mt/k), m) = 1$, where k denotes the multiplicative order of 2 modulo p . In the GNB representation, from $\gamma^p = 1$, the field element B can be rewritten as

$$B = b_{F(1)}\gamma + b_{F(2)}\gamma^2 + \cdots + b_{F(p-1)}\gamma^{p-1} \quad (3)$$

where $F(2^i 2^m j \bmod p) = i$, for $0 \leq i \leq m-1$ and $0 \leq j \leq t-1$ and, αB can be obtained as

$$\alpha B = \sum_{i=0}^{t-1} B^{(\langle 2^i m \rangle_p)} \quad (4)$$

where $B^{(i)} = \sum_{j=0}^{p-1} b_{F(j-i)}\gamma^j$, and αB in (4) can be translated from the GNB into NB form $\alpha B = \sum_{i=0}^{m-1} \bar{b}_i \alpha^{2^i}$. Each term $\bar{b}_i$ is computed by

$$\bar{b}_i = \begin{cases} \sum_{j=0}^{t-1} b_{F(2^i - 2^m j)} + b_{F(0)}, & \text{if } t = \text{odd} \\ \sum_{j=0}^{t-1} b_{F(2^i - 2^m j)}, & \text{if } t = \text{even} \end{cases} \quad (5)$$

which satisfy the following properties.

Proposition 1: If t is even, $\bar{b}_i = \sum_{j=0}^{t-1} b_{F(2^i - 2^m j)}$ for $0 \leq i \leq m-1$ have t terms of b_i , $1 \leq i \leq m-1$, and $(t-1)$ terms of b_0 .

Proposition 2: If t is odd number, $\bar{b}_i = \sum_{j=0}^{t-1} b_{F(2^i - 2^m j)} + b_{F(0)}$ for $0 \leq i \leq m-1$ have t terms of b_i , with $i \neq m/2$ and $(t-1)$ terms of $b_{m/2}$, where $b_{F(0)} = b_{m/2}$.

Using these propositions, the parity prediction of $\alpha B^{2^{-(m-i)}}$ can, in general, be found to be

$$\hat{P}_{\alpha B^{2^{-(m-i)}}} = \begin{cases} b_i, & \text{for } t = \text{even} \\ P_B + b_{\langle (m/2) - i \rangle}, & \text{for } t = \text{odd}. \end{cases} \quad (6)$$

B. Parity Prediction of PS Module

The square of an element A in NB is given by $A^2 = (a_{m-2}, a_0, \dots, a_{m-2})$, and it is easy to show that the parity prediction of A^2 equals to $P_{A^2} = P_A$. The parity prediction of scalar multiplication gA , where $g \in \text{GP}(2)$, is equivalent to $P_{gA} = g \cdot P_A$. The parity prediction of $\alpha B^{2^{-(m-i)}}$ is derived in the previous subsection, and computation of $\alpha B^{2^{-(m-i)}}$ is performed in the α module. Thus, the parity prediction of C in (2) can be obtained as

$$\hat{P}_{C_i} = \hat{P}_{C_i} + a_{m-i} \hat{P}_{\alpha B^{2^{-(m-i)}}} \quad (7)$$

C. Proposed CED Architecture

We have derived two individual modules for determination of parity prediction bits $\hat{P}_{\alpha B^{2^{-(m-i)}}}$ and $\hat{P}_{C_i}$ of $\alpha B^{2^{-(m-i)}}$ and C_i , respectively. The proposed NB multiplier over $GF(2^m)$ with CED is shown in Fig. 3. Let us consider the case where a stuck-at fault occurs at any of the XOR or AND gates in α' and PS' modules (as shown in Fig. 4). This error can be detected by $\hat{e}_{C_i}$, where $\hat{e}_{C_i}$ is the output of XOR gate. The parity prediction bit $\hat{P}_{C_i}$ is compared with the actual parity bit P_{C_i} , where the indicator $\hat{e}_{C_i} = \hat{P}_{C_i} + P_{C_i}$ is the presence of a single fault in α and PS modules. The i th intermediate result is fault-free if $\hat{e}_{C_i} = 0$. Note that, in α' and PS' modules, all result bits are independent. At the i th computational loop, if fault occurs at the i th bit, that would lead to permanent fault in the output of in α' or PS' modules, i.e., the error is not injected on other output bits. After the i th loop, the result is subjected to cyclic shifting operation and then added to the results of the $(i + 1)$ th computational loop. Thus, the result in the $(i + 1)$ th loop contains even number of errors, and the results in the $(i + 2)$ th loop contains odd number of errors. In Fig. 3, the error detection operations are performed in each computational loop. Therefore, the errors due to any single permanent fault in the proposed CED multiplier can be detected. When the odd number of permanent faults exist in the output of in α' and PS' modules, the presence of errors is the same of a single permanent fault.

IV. FAULT DETECTION ARCHITECTURE IN BIT-SERIAL NORMAL BASIS MULTIPLIER USING MULTIPLE PARITY PREDICTION SCHEMES

Assume that the element B is split into $n = \lceil m/d \rceil$ sub-words, where d is the selected digit size. The element B in NB then can be represented by

$$B = B_0 + B_1^{2^d} + \dots + B_{n-1}^{2^{d(n-1)}} \quad (8)$$

where $B_i = b_{id}\alpha + b_{id}\alpha^2 + \dots + b_{id+d-1}\alpha^{2^{d-1}}$. It is clear that the element B is also a square structure of the sub-word B_i . Therefore, from (5) and (6), αB can be obtained as

$$\begin{aligned} \alpha B &= \alpha B_0 + \alpha B_1^{2^d} + \dots + \alpha B_{n-1}^{2^{d(n-1)}} \\ &= \bar{B}_0 + \bar{B}_1^{2^d} + \dots + \bar{B}_{n-1}^{2^{d(n-1)}} \end{aligned} \quad (9)$$

where $\bar{B}_i = \bar{b}_{id}\alpha + \bar{b}_{id}\alpha^2 + \dots + \bar{b}_{id+d-1}\alpha^{2^{d-1}}$.

In the i th loop of computation in (2), the intermediate result is calculated by $C_i = C_{i-1}^2 + \alpha_{m-i}\alpha B^{2^{-(m-i)}}$. When C_{i-1}^2 and αB are can now be sliced into $C_{i-1}^2 = \bar{C}_{i-1,0} + \bar{C}_{i-1,1}^{2^d} + \dots +$

$\bar{C}_{i-1,n-1}^{2^{d(n-1)}}$ and $\alpha B = \bar{B}_0 + \bar{B}_1^{2^d} + \dots + \bar{B}_{n-1}^{2^{d(n-1)}}$, respectively, so that C_i can be computed as $C_i = \sum_{j=0}^{n-1} C_{i,j}^{2^{dj}}$, where $C_{i,j} = \bar{C}_{i-1,j} + a_{m-i}\bar{B}_j$. From (9), we can obtain the parity prediction of $\bar{B}_j = \alpha B_j$ as

$$P_{\bar{B}_j} = \bar{b}_{jd} + \bar{b}_{jd+1} + \dots + \bar{b}_{jd+d-1} \quad (10)$$

where each coefficient $\bar{b}_j$ is defined in (5).

Moreover, since $C_{i-1}^2 = \bar{C}_{i-1,0} + \bar{C}_{i-1,1}^{2^d} + \dots + \bar{C}_{i-1,n-1}^{2^{d(n-1)}}$ each term $\bar{C}_{i-1,j}$ is

$\bar{C}_{i-1,j} = C_{i-1,j} + c_{i-1,jd-1} + c_{i-1,d(j+1)}$. Thus, the parity prediction of $\bar{C}_{i-1,j}$ can be computed by

$$\hat{P}_{\bar{C}_{i-1,j}} = \hat{P}_{C_{i-1,j}} + c_{i-1,jd-1} + c_{i-1,d(j+1)}. \quad (11)$$

Applying (10) and (11), the parity prediction of $C_{i,j} = \bar{C}_{i-1,j} + a_{m-i}\bar{B}_j$ can be calculated as

$$\begin{aligned} \hat{P}_{C_{i,j}} &= \hat{P}_{\bar{C}_{i-1,j}} + a_{m-i}\hat{P}_{\bar{B}_j} \\ &= \hat{P}_{C_{i-1,j}} + c_{i-1,jd-1} + c_{i-1,d(j+1)} + a_{m-i}\hat{P}_{\bar{B}_j}. \end{aligned} \quad (12)$$

Note that the predicted parity bit $\hat{P}_{\bar{B}_j}$ is estimated for the output of αB_j . Fig. 5 shows the proposed bit-serial normal basis multiplier based on the proposed multiple parity prediction schemes using (10) and (12). Fig. 6 shows the structure of PS'_j module. Each of the PS'_j module uses three additional XOR gates and one additional AND gate to generate $\hat{P}_{C_{i,j}}$ according to (12). This circuit produces n indicator bits $\hat{e}c_{i,j}$ to detect the intermediate output of each loop of computation.

The actual parity bit $P_{C_{i,j}}$ is added with the predicted parity bit $\hat{P}_{C_{i,j}}$ to obtain the indicator $\hat{e}c_{i,j}$ (i.e., $\hat{e}c_{i,j} = \hat{P}_{C_{i,j}} + P_{C_{i,j}}$) such that $\hat{e}c_{i,j}$ is 0 when the predicted parity is equal to the actual parity and 1 otherwise. The structure for determination of the actual parity bit $P_{C_{i,j}}$ needs $(d-1)$ XOR gates, and its computational delay amounts to $\lceil \log_2 d \rceil$ XOR gate-delays. Note that the proposed structure of computing the predicted parity bit $\hat{P}_{C_{i,j}}$ does not depend on the original normal multiplier. Consequently, as the original normal multiplier is affected by stuck-at faults, the errors are not injected into the circuit for the computation of predicted parity bit. From the structure of Fig. 5, the core processing circuit is divided by n components to perform each computational loop. Since a single stuck-at fault model is assumed in each component, a single fault in each component produces odd number of errors; the total number of effective errors is then either odd or even. If one of the sub-modules in the system with the even number of permanent faults occurs, then the structure in Fig. 5 cannot detect the errors. After each computation in the loop, the result is shifted by one bit to add the result of next computation. The errors can be detected even when one of errors is shifted to neighboring sub-module. Therefore, the structure in Fig. 5 can detect multiple errors in the intermediate results computed by the loop.

V. ANALYSIS PERFORMANCE

A. Probability of Error Detection

Under the single stuck-at fault model (assumed in this paper), α or PS modules of Fig. 4 is incurred by the fault injection which is like the combinational logic errors. In the proposed CED scheme of Fig. 3, we assume that p is the probability of an error in any computational loop, and E_i , denotes the probability of an undetected error in the i th computational loop. Thus, we can obtain the following relation:

$$E_i = pO_{i-1} + (1 - p)E_{i-1} = (1 - 2p)E_{i-1} + p \quad (13)$$

where $O_i (= 1 - E_i)$ is the probability of a detected error in the i th computational loop. Using (13), we can have E_i , as

$$\begin{aligned} E_i &= (1 - 2p)E_{i-1} + p \\ &= (1 - 2p)^2 E_{i-2} + p(1 - 2p) + p \\ &= \dots \\ &= (1 - 2p)^i E_0 + \sum_{k=0}^{i-1} p(1 - 2p)^k. \end{aligned} \quad (14)$$

$E_0=0$, since the input data is correct, and thus we have

$$\begin{aligned} E_i &= (1 - 2p)^i + \sum_{k=0}^{i-1} p(1 - 2p)^k \\ &= (1 - 2p)^i + p \frac{(1 - 2p)^i - 1}{(1 - 2p) - 1} = \frac{(1 - 2p)^i + 1}{2}. \end{aligned} \quad (15)$$

Since the estimated E_i in (15) has zero as inputs with the probability $(1 - p)^i$. Therefore, E_i , needs to be modified as

$$E_i = \frac{(1 - 2p)^i + 1}{2} - (1 - p)^i. \quad (16)$$

Considering the behavior of error detection in every computational loop, the probability of undetected errors can be calculated by

$$Pr(e) = \prod_{i=0}^{m-1} E_i. \quad (17)$$

Thus, the probability of detected errors in Fig.3 is

$$\Pr_D(e) = 1 - \Pr_U(e). \quad (18)$$

For multiple parity prediction scheme, the probability of detected errors in Fig. 5 with n parity bits is given by

$$\Pr_D(e) = 1 - \left(\Pr_U(e) \right)^n. \quad (19)$$

Using MATLAB 6.5 we have simulated the proposed multipliers over $GF(2^{30})$ with 10 000 random inputs, by fault injection with probability $p = 0.03$, $p = 0.1$, and $p = 0.2$. The simulation results in terms of probability versus number of loops are shown in Fig. 7. When $p = 0.2$, the probability of error detection for parity bits $n = 1, 2, 3$ and 4 is nearly 99% after 16th, 11th, 8th, 7th computational loops, respectively. In Fig. 7, we have plotted the probability of detection at different computational loops obtained from the simulation. The detection probability obtained from the simulation is nearly the same the detection probability given by (18) and (19). According to the simulation results, the proposed CED NB multipliers have about 100% the probability of error detection. We have estimated the detection probability of Fenn's multiplier [6] according to (16) and found to have nearly 50% error detection capability.

B. Analysis Time, Area, and Area $\times$ Time Overheads

From the structure of CED NB multiplier in Fig. 3, the space overhead of the proposed circuit for all-types of normal basis multipliers is found to be $(m+2)$ XOR gates, one AND gate and one 1-bit latch. Duration of each cycle amounts to $[\log_2(m+1)] T_{XOR} + T_L$ gate delays. For estimation of complexity and comparison, we can find the area and delays of the AND gates, XOR gates and latches as follows : $T_{AND} = 1.5\Delta$, $A_{AND} = 1.8A_u$; $T_{XOR} = 1.7\Delta$, $A_{XOR} = 2 A_u$; $T_L = 6\Delta$, $A_L = 5.75 A_u$, where the time unit Δ , and the area unit A_u , are the delay and area of an inverter circuit. As discussed in Section III-A, in $GF(2^{233})$ type- t NBs exists for a given t if $p = mt + 1$ is prime (such as $t = 2, 6, 12$ and 30). Table I shows area, time and area-time overheads in the type- t bit-serial NB multipliers over $GF(2^{233})$. It is shown that the area and time complexity of the NB multipliers decreases with t . The overheads corresponding to different number of parity-bits in bit-serial type-2 NB multiplier over $GF(2^{233})$ using multiple parity prediction scheme is shown in Table II. We can find that, while the time overhead decreases, the space overhead increases with the number of parity-bits. When the number of parity bits is eight, the proposed multiplier (in Fig. 5) has about 5% area-time overhead. In Table III we have compared the area and time complexities of the proposed CED multipliers with the existing CED NB multiplier [6]. The area and delay complexities of bit- serial type-2 NB multiplier over $GF(2^9)$ with and without parity prediction schemes are determined by the Synopsys Design Compiler using 0.18- μm TSMC library [14]. The simulation results are listed in Table IV, and found to be matching with the theoretical estimation.

VI. CONCLUSION

Using the structure of a bit-by-byte NB multiplier [11], we have presented parity prediction schemes to detect errors in bit-serial NB multipliers using single and multiple parity-bits.

Although different types of NB multipliers have different time and space complexities, it is shown that all types of NB multipliers involve the same structure for the implementation of parity prediction scheme. As the value of t increases, the type- t NB multipliers with CED capability involve lower space overhead for architectural modification. The probability of error detection by the proposed structure with single parity-bit is about 100%, while Fenn's multiplier [6] is 50%. By multiple parity prediction approach, we have proposed an NB multiplier with CED capability, which has $\sim 5\%$ area-time overhead. It has error detection capability with nearly 99% probability after 8th computational loop when three parity bits are used for error prediction.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable comments which significantly improved the revised version of this paper.

REFERENCES

- [1] K. H. Huang and J. A. Abraham, "Algorithm-based fault tolerance for matrix operations," *IEEE Trans. Computers*, vol. 33, no. 6, pp. 518–522, Jun. 1984.
- [2] D. P. Agrawal and V. K. Agarwal, "On-line bus fault diagnosis in microprocessor systems," *J. Digital Syst.*, vol. 4, pp. 337–391, 1980.
- [3] C. Y. Lee, C. W. Chiou, and J. M. Lin, "Concurrent error detection in polynomial basis multiplier over $GF(2^m)$," *J. Electron. Test.*, vol. 22, pp. 143–150, Feb. 2006.
- [4] C. W. Chiou, C. Y. Lee, A. W. Deng, and J. M. Lin, "Concurrent error detection in montgomery multiplication over $GF(2^m)$," *IEICE Trans. Fund.*, vol. E89-A, no. 2, pp. 566–574, Feb. 2006.
- [5] C. Y. Lee, "Concurrent error detection in digit-serial normal basis multiplication over $GF(2^m)$," in *22st Intern. Conf. Advanced Information Networking and Applications Workshops*, 2008, pp. 1499–1504.
- [6] S. Fenn, M. Gossel, M. Benaissa, and D. Taylor, "On-line error detection for bit-serial multipliers in $GF(2^m)$," *J. Electron. Test.*, vol. 13, pp. 29–40, 1998.
- [7] A. Reyhani-Masoleh and M. A. Hasan, "Fault detection architectures for field multiplication using polynomial bases," *IEEE Trans. Comp.*, vol. 55, no. 9, pp. 1089–1103, Sep. 2006.
- [8] S. Bayat-Sarmadi and M. A. Hasan, "On concurrent detection of errors in polynomial basis multiplication," *IEEE Trans. VLSI Syst.*, vol. 15, pp. 413–426, 2007.
- [9] W. Chelton and M. Benaissa, "Concurrent error detection in $GF(2^m)$ multiplication and its application in elliptic curve cryptography," *IET Circuits Devices Syst.*, vol. 2, no. 3, pp. 289–297, 2008.
- [10] C. Y. Lee, C. W. Chiou, and J. M. Lin, "Concurrent error detection in a bit-parallel systolic multiplier for dual basis of $GF(2^m)$," *J. Electron. Test.*, vol. 21, pp. 539–549, Sep. 2005.
- [11] T. Beth and D. G. Mann, "Algorithm engineering for public key algorithms," *IEEE J. Selected Areas in Comm.*, vol. 7, no. 4, pp. 458–465, May 1989.
- [12] S. Feisel, J. V. Z. Gathen, and M. Shokrollahi, "Normal bases via general Gauss periods," *Math. Comp.*, vol. 68, pp. 271–290, 1999.
- [13] S. Gorshe and B. Bose, "A self-checking ALU design with efficient codes," in *Proc. IEEE 14th VLSI Test Symp '96*, pp. 157–161.
- [14] Synopsys, DesignWare Foundry Libraries, Mountain View, CA [Online]. Available: <http://www.synopsys.com/>

LIST OF FIGURES

- Figure 1 Bit-serial normal basis multiplier over $GF(2^m)$.
- Figure 2 Multiplier with concurrent error detection capability.
- Figure 3 Proposed bit-serial normal basis multiplier for CED
- Figure 4 (a) Detail circuit of the PS' module. (b) Detail circuit of the α' module for $t = \text{even}$.
- Figure 5 Proposed bit-serial NB multiplier using multiple parity for CED.
- Figure 6 Detail circuit of the PS'_j module.
- Figure 7 Probability versus computational loops.

LIST OF TABLES

Table 1	Area, time and area-time overheads in bit-serial type- t multipliers over GF (2^{333}).
Table 2	Area and time overheads in bit-serial type-2 NB multiplier over GF(2^{333}) using multiplier parity prediction.
Table 3	Comparison of bit-serial NB multipliers using parity prediction scheme.
Table 4	Simulation results of area and delay complexities of multipliers with/without parity prediction for $m=9$.

Figure 1

Figure 2

Figure 3

Figure 4

Figure 5

Figure 6

Figure 7

t	2	6	12	30
time overhead (%)	79.8	37	22.5	10.7
area overhead (%)	11.9	8	5.47	2.78
area-time overhead (%)	9.5	2.9	1.23	0.29

Table 1

parity bits	2	4	6	8	10	12
time overhead (%)	65.7	49.7	33.8	33.8	33.8	17.8
area overhead (%)	12.5	13.3	14.2	15	15.7	16.5
area-time overhead (%)	8.2	6.5	4.8	5	5.3	2.9

Table 2

multipliers	Fenn et al.[6]	Fig.3	Fig.5
# parity bits	1	1	n bits
error detection probability	50 %	100 %	100 %
time overhead	$T_A + kT_X + T_L$	$\lceil \log_2 m \rceil T_X$	$\lceil \log_2 d \rceil T_X$
area overhead			
#XOR	$t + 3$	$m + 2$	$(5 + d)n$
#AND	2	1	n
#Latch	5	1	n

note: $n = \lceil m/d \rceil$ and $k = \lceil \log_2 t \rceil + 1$

Table 3

multipliers design	area (sq.um)	delay (ns)	% overheads	
			area	time
without parity prediction	2604.6	6.11		
with single parity-bit	3003.7	14.6	13.3	58.2
with 3 parity-bits (multiple-parity)	3569.2	14.3	27.0	57.3

Table 4