Low-hardware Digit-serial Sequential Polynomial Basis Finite Field GF(2^m) Multiplier for Trinomials

Siva Ramakrishna Pillutla and Lakshmi Boppana

National Institute of Technology Warangal, Telangana, India
srk100p@student.nitw.ac.in

Abstract. Finite field GF(2^m) multipliers are employed in practical applications such as elliptic curve cryptography (ECC) and Reed-Solomon encoders. Digit-level finite field multipliers are best suitable for applications that require low hardware implementation while operating at speeds that conforms to today's high data rates. With the emergence of Internet of Things (IoT), many resource-constrained devices such as IoT edge devices came into proliferate usage. To secure these constrained devices, ECC must be implemented in these devices with low hardware complexity. Hence, it requires to design efficient digit-serial finite field multipliers since the performance of ECC greatly depends on the performance of the finite field multiplier employed. Many efficient designs for digit-serial finite field multipliers are presented in the literature to achieve better area and time complexities. In this paper, we present an area-efficient sequential digit-serial finite field multiplier for trinomials. The hardware and time complexities of the proposed multiplier are estimated for GF(2^{409}) and compared with the similar multipliers available in the literature. The comparison shows that the proposed multiplier achieves lower hardware complexity. Therefore, the proposed multiplier is attractive for cost-effective high-speed applications such as IoT edge devices.

Keywords: Internet of Things · Elliptic curve cryptography · Digit-level multiplier · Finite field arithmetic.

1 Introduction

Internet of Things (IoT) connects many physical things that are quite different from conventional computing systems to the network. These physical things are equipped with constrained devices, namely, IoT end-devices/IoT edge-devices, to enable them to participate in communication over the network. These constrained IoT devices must be cost-effective while suitable for today's high data speed applications. Enabling security in IoT edge-devices is crucial to thwarting many network-based attacks [3]. Some of the security features can be achieved by using public key cryptosystems such as elliptic curve cryptography (ECC) and RSA (Rivest-Shamir-Adleman). ECC with its relatively shorter key-size and relatively less computational requirements is best suitable for constrained devices

(IoT edge devices) to implement security services such as digital signature and key-exchange [8].

Since arithmetic operations in ECC heavily involve over finite fields GF(2^m), efficient finite field arithmetic implementations, particularly multiplication, results in a considerable performance improvement in ECC [4]. Hence, the performance of applications that employs ECC for security implementation can be enhanced by designing an efficient finite field multiplier. For IoT edge devices, designing a bit-serial multiplier results in low cost. However, this multiplier is too slow for today's high data speed applications. Designing a parallel multiplier requires very high hardware that results in more cost, which is not desirable for IoT end-devices/IoT edge-devices since a typical domestic application needs tens of IoT devices. Digit-serial multipliers inherently provide area and delay trade-off and can also make use of full available data bus width of an IoT device are best suitable for IoT edge-devices.

A finite field GF(2^m) has 2^m elements, where the field elements can be represented using various bases such as dual basis, normal basis, polynomial basis, and redundant basis. Polynomial basis is one of the bases recommended by many standard institutes such as National Institute of Standards and Technology (NIST), and designs based on this basis are more regular and modular. In polynomial basis representation, every field element is a reduced modulo an irreducible polynomial. There are various types of irreducible polynomials such as trinomials, pentanomials, all one polynomials, and general irreducible polynomials. Standard institutes recommend sparse polynomials such as trinomials and pentanomials as they result in low hardware and time complexities. Hence, fields defined over trinomials are more suitable for constrained devices.

Many digit-level finite field multipliers over trinomials are proposed in the literature to achieve better area and time complexities. In [7], two digit-level multipliers based on most significant bit (MSB)/least significant bit (LSB) first algorithms were presented. In this paper, the authors have shown that irreducible polynomials that have low hamming weight and low second highest degree result in complexity reduction. In [10], a bit-parallel word serial multiplier for GF(2^{233}) over trinomials was presented. This multiplier has a parallel partial product generator followed by an accumulation unit. In [5], a high-throughput hardware efficient digit-serial architecture was presented. In this multiplier, by using T flip flops in the accumulation unit authors achieved lower critical path delay as well as low hardware complexity. In [2], a shifted polynomial basis (SPB) digit-serial multiplier using the proposed (b, 2)-way Karatsuba decomposition was presented to achieve sub-quadratic space complexity. A digit-serial area-efficient multiplier employing a new factoring technique was proposed in [6] to achieve power reduction. In our paper, we propose a new area-efficient polynomial basis digit-level multiplier whose structure comprises of a parallel multiplier followed by accumulation unit. The parallel multiplier is based on the approach proposed in [9] for a parallel multiplier for all trinomials. This approach when applied for a class of trinomials, $x^m + x^n + 1$, for which $n \leq m/2$ gives low hardware implementation. The class of trinomials considered in this paper also includes

two of the five NIST recommended irreducible polynomials suggested for ECC. The proposed area-efficient digit-serial multiplier is suitable for edge devices used in IoT applications.

The organization of the paper is as follows. Section 2 gives preliminaries regarding polynomial basis digit-serial multiplication. Section 3 introduces the mathematical formulations for the proposed digit-serial multiplication and presents the proposed architecture. In addition, comparison of area and time complexities of the proposed multiplier with the existing similar multipliers is also presented in this section. Conclusions are presented in Section 4.

2 Preliminaries

A finite field GF(2^m) has 2^m elements where each element is represented with a polynomial of degree at most $(m-1)$ over GF(2). An m^{th} degree polynomial over which the field GF(2^m) is defined is called the irreducible polynomial $P(x)$ of the field, given by $P(x) = x^m + \sum_{j=m-1}^{1} p_j x^j + 1$, where all the $p_j \in$ GF(2). The polynomial basis can be defined with the set $(x^{m-1}, x^{m-2}, ..., x^2, x, 1)$, where x is the root of the irreducible polynomial $P(x)$ of the field.

Let A and B be two arbitrary field elements, given by

$$\begin{aligned} A(x) &= \sum_{j=0}^{m-1} a_j x^j \\ &= a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + ... + a_1 x^1 + a_0 \end{aligned} \tag{1}$$

and

$$\begin{aligned} B(x) &= \sum_{j=0}^{m-1} b_j x^j \\ &= b_{m-1}x^{m-1} + b_{m-2}x^{m-2} + ... + b_1 x^1 + b_0, \end{aligned} \tag{2}$$

where all a_j and $b_j \in$ GF(2). Let $D(x)$ be the product of $A(x)$ and $B(x)$. Conventionally, $D(x)$ is obtained by first multiplying the two polynomials A and B followed by modulo reduction using irreducible polynomial $P(x)$. Thus,

$$D(x) = A(x)B(x) \bmod P(x) \tag{3}$$

Two popular digit-level schemes presented in the literature to evaluate $D(x)$ are MSD (most significant digit) first scheme and LSD (least significant digit) first scheme. Let s be the number of digits and l be the width of each digit, then operand B can be written as

$$B = \sum_{j=0}^{s-1} B_j x^{jl} \tag{4}$$

where

$$B_j = \begin{cases} \sum_{t=0}^{l-1} b_{lj+t} x^t, & \text{for } 0 \le j \le s-2 \\ \sum_{t=0}^{m-1-l(s-1)} b_{lj+t} x^t, & \text{for } j = s-1 \end{cases}$$

and $s = \lceil \frac{m}{l} \rceil$. Then, $D(x)$ can be obtained as

$$D = (B_0 A + B_1(Ax^l \bmod P(x)) + B_2(Ax^l x^l \bmod P(x)) + ...+ B_{s-1}(Ax^{l(s-2)} x^l \bmod P(x))) \bmod P(x) \quad (5)$$

for the LSD scheme, and

$$D = ((...(((AB_{s-1} \bmod P(x))x^l + AB_{s-2}) \bmod P(x))x^l + ...)x^l + AB_0) \bmod P(x) \quad (6)$$

for the MSD scheme.

3 Proposed digit-serial multiplier

3.1 Mathematical Formulation

Let $P(x) = x^m + x^n + 1$, where $1 \le n \le \lceil \frac{m}{2} \rceil$, be an irreducible trinomial polynomial over which the field GF(2^m) is defined. Let $A(x) = \sum_{j=0}^{m-1} a_j x^j$ and $B^{'}(x) = \sum_{j=0}^{l-1} b_j^{'} x^j$ be two elements, where $l \le m$. Let $C(x)$ denote the product of polynomials A and $B^{'}$ as $C(x) = \sum_{j=0}^{m+l-2} c_j x^j = AB^{'}$. This product expression $C(x) = AB^{'}$ can be expressed using a $(m + l - 1) \times l$ matrix M as follows.

$$\begin{bmatrix} c_0 \\ c_1 \\ . \\ . \\ . \\ c_{l-1} \\ c_l \\ . \\ . \\ . \\ c_{m-1} \\ c_m \\ . \\ . \\ . \\ c_{m+l-2} \end{bmatrix} = \begin{bmatrix} a_0 & 0 & 0 & .... & 0 & 0 \\ a_1 & a_0 & 0 & .... & 0 & 0 \\ . & . & . & .... & . & . \\ . & . & . & .... & . & . \\ . & . & . & .... & . & . \\ a_{l-1} & a_{l-2} & a_{l-3} & .... & a_1 & a_0 \\ a_l & a_{l-1} & a_{l-2} & .... & a_2 & a_1 \\ . & . & . & .... & . & . \\ . & . & . & .... & . & . \\ . & . & . & .... & . & . \\ a_{m-1} & a_{m-2} & a_{m-3} & .... & a_{m-l+1} & a_{m-l} \\ 0 & a_{m-1} & a_{m-2} & .... & a_{m-l+2} & a_{m-l+1} \\ . & . & . & .... & . & . \\ . & . & . & .... & . & . \\ . & . & . & .... & . & . \\ 0 & 0 & 0 & .... & 0 & a_{m-1} \end{bmatrix} \times \begin{bmatrix} b_0^{'} \\ b_1^{'} \\ b_2^{'} \\ . \\ . \\ . \\ b_{l-1}^{'} \end{bmatrix}$$

The product polynomial $C(x)$ includes the terms whose degree is more than $m-1$. These terms can be modulo reduced using the identity $x^m = x^n + 1$. From this identity we can have $x^{m+i} = (x^n + 1)x^i = x^{n+i} + x^i$, where $0 \leq i \leq (l-2)$. Assume $l \leq \lceil m/2 \rceil$, then we have $n + i \leq n + l - 2 \leq m/2 + \lceil m/2 \rceil - 2 < m$. Thus each term in the product polynomial $C(x)$ whose degree is $(m+i)$ can be reduced to a polynomial of at most degree $(m-1)$ with two terms, $x^{n+i} + x^i$. By this modulo reduction, each $(m+i)^{th}$ row of M for $0 \leq i \leq (l-2)$ is added to the i^{th} and $(n+i)^{th}$ rows of M.

Let Q be the $m \times l$ matrix obtained from the matrix M, after the reduction process applied. Let matrix Q be decomposed into the sum of three $m \times l$ matrices, X, Y, and Z, as $Q = X + Y + Z$. The three matrices X, Y, and Z can be defined as follows.

$$X = \begin{bmatrix} a_0 & 0 & 0 & .... & 0 & 0 \\ a_1 & a_0 & 0 & .... & 0 & 0 \\ . & . & . & .... & . & . \\ . & . & . & .... & . & . \\ . & . & . & .... & . & . \\ a_{l-2} & a_{l-3} & a_{l-4} & .... & a_0 & 0 \\ a_{l-1} & a_{l-2} & a_{l-3} & .... & a_1 & a_0 \\ . & . & . & .... & . & . \\ . & . & . & .... & . & . \\ . & . & . & .... & . & . \\ a_{m-1} & a_{m-2} & a_{m-3} & .... & a_{m-l+1} & a_{m-l} \end{bmatrix}$$

$$Y = \begin{bmatrix} 0 & a_{m-1} & a_{m-2} & \dots & a_{m-l+2} & a_{m-l+1} \\ 0 & 0 & a_{m-1} & \dots & a_{m-l+3} & a_{m-l+2} \\ . & . & . & \dots & . & . \\ . & . & . & \dots & . & . \\ . & . & . & \dots & . & . \\ 0 & 0 & 0 & \dots & 0 & a_{m-1} \\ 0 & 0 & 0 & \dots & 0 & 0 \\ . & . & . & \dots & . & . \\ . & . & . & \dots & . & . \\ . & . & . & \dots & . & . \\ 0 & 0 & 0 & \dots & 0 & 0 \end{bmatrix}$$

$$Z = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ . & . & . & \dots & . & . \\ . & . & . & \dots & . & . \\ 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & a_{m-1} & a_{m-2} & \dots & a_{m-l+2} & a_{m-l+1} \\ 0 & 0 & a_{m-1} & \dots & a_{m-l+3} & a_{m-l+2} \\ . & . & . & \dots & . & . \\ . & . & . & \dots & . & . \\ 0 & 0 & 0 & \dots & 0 & a_{m-1} \\ . & . & . & \dots & . & . \\ . & . & . & \dots & . & . \\ 0 & 0 & 0 & \dots & 0 & 0 \end{bmatrix} \begin{matrix} \\ \\ \\ 0^{th} row \\ \\ n^{th} row \\ \\ \\ \\ \\ (n+l-2)^{th} row \\ \\ \\ \end{matrix}$$

Matrix Z is equivalent to a matrix that can be obtained by shifting matrix Y down by n rows and filling the first n rows with zeros. By employing similar method presented in [9], any i^{th} row of matrix Q can be obtained with simple rewiring of the n^{th} row, Q_n, of the matrx Q. The row Q_n can be computed as

$$Q_n = \begin{cases} (a_n a_{n-1} ... a_0 a_{m-1} a_{m-2} .... a_{m-l+n+1}) + (0 a_{m-1} ... a_{m-l+1}), & \text{if } n \leq l \\ (a_n a_{n-1} ..... a_{n-l+1}) + (0 a_{m-1} ... a_{m-l+1}), & \text{if } n > l \end{cases}$$

To compute Q_n, it requires $(l-1)$ two-input XOR gates and a delay of T_X, where T_X is a dealy of a two-input XOR gate. Since Q is an $m \times l$ matrix, $Q.B^{'}$ requires lm AND gates, $(l-1)m$ XOR gates and $T_A + \lceil log_2^l \rceil T_X$ delays, where $B^{'} = (b_0, b_1, ..., b_{l-1})^t$.

Let $A(x) = \sum_{i=0}^{m-1} a_i x^i$ and $B(x) = \sum_{i=0}^{m-1} b_i x^i$ be two arbitrary field elements. Let $D(x) = \sum_{i=0}^{m-1} d_i x^i = AB \bmod P(x)$ be the product of the elements A and B. The computation of $D(x)$ can be performed as follows.

Let the element $B(x)$ is partitioned into s digits where each digit of size l bits. Then, we have $s = \lceil \frac{m}{l} \rceil$. It follows,

$$\begin{aligned} B(x) &= \sum_{j=0}^{l-1} b_j x^j + \sum_{j=l}^{2l-1} b_j x^j + ... + \sum_{j=(s-1)l}^{sl-1} b_j x^j \\ &= \sum_{j=0}^{l-1} b_j x^j + x^l \sum_{j=0}^{l-1} b_{l+j} x^j + +x^{2l} \sum_{j=0}^{l-1} b_{2l+j} x^j ..... + x^{(s-1)l} \sum_{j=0}^{l-1} b_{(s-1)l+j} x^j, \end{aligned} \tag{7}$$

where all b_js for $j \geq m$ are zero. Now, the product $D(x)$ can be computed as

$$\begin{aligned} D(x) = A(x)B(x) \bmod P(x) &= A(x) \sum_{j=0}^{l-1} b_j x^j \bmod P(x) + \\ x^l A(x) \sum_{j=0}^{l-1} b_{l+j} x^j \bmod P(x) &+ x^{2l} A(x) \sum_{j=0}^{l-1} b_{2l+j} x^j \bmod P(x) + ... \\ &..... + x^{(s-1)l} A(x) \sum_{j=0}^{l-1} b_{(s-1)l+j} x^j \bmod P(x) \end{aligned}$$

$$\begin{aligned} = T_0 \bmod P(x) + x^l T_1 \bmod P(x) + x^{2l} T_2 \bmod P(x) + ... \\ .... + x^{(s-1)l} T_{s-1} \bmod P(x) \end{aligned}$$

$$\begin{aligned} = (.....((T_{s-1} x^l \bmod P(x) + T_{s-2}) x^l \bmod P(x) + T_{s-3}) x^l \bmod P(x) + ... \\ ..... + T_1) x^l \bmod P(x) + T_0, \end{aligned} \tag{8}$$

where

$$T_i = A(x) \sum_{j=0}^{l-1} b_{il+j} x^j \bmod P(x). \tag{9}$$

The computation of T_i can be performed using the procedure shown to compute $C(x)$. In the computation of $C(x)$, note that the value of digit-size, l, is taken at most half the value of field order, m. It is acceptable for the constrained devices since the data bus width of these devices is typically 8/16/32 bits only. As per today's security requirements, a field order of at least 233 is required. Hence, the selected l range is quite applicable to today's security requirements for Wireless Sensor Network (WSN) nodes and IoT end-nodes/edge-devices.

3.2 Proposed Structure of the Multiplier

Based on the proposed formulations, a conceptual block diagram of the digit-serial multiplier is shown in Figure 1. The structure shown in Figure 1 realizes the expression given in Equation 8. Node G1 is a partial parallel $m \times l$ multiplier that multiplies an m-bit element with an l-bit element. It realizes the computation of T_i as given in Equation 9. Node $G2$ performs the additions that are involved in the computation of the expression in Equation 8. Similarly, Node $G3$ performs

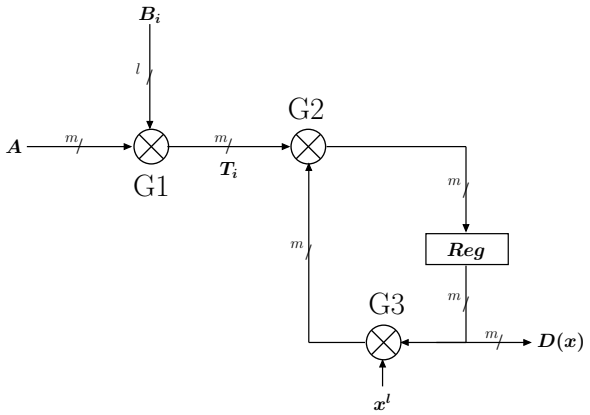

Fig. 1. The Proposed structure of the digit-serial multiplier.

the interleaved multiplications of partial output product with x^l that are involved in the computation of the expression in Equation 8. The multiplicand A is made available throughout the computation, while multiplier B enters the structure digit-wise starting from most significant digit (MSD). The structure produces the required multiplication result after a dealy of s clock cycles.

3.3 Area and Time Complexities

In this section, the area and time complexities of the proposed multiplier are obtained and compared with the existing similar multipliers. The node $G1$ which computes T_i performs a similar computation presented for computing $Q.B^{'}$. Hence, it requires lm AND gates and $(l-1)(m+1)$ XOR gates. The node $G2$ requires m XOR gates while the node $G3$ requires l XOR gates. The structure also requires two m-bit registers, one at the input to register multiplicand A while another as output register, *Reg*. The delays of the nodes $G1, G2$, and $G3$ are $T_A + (\lceil log_2^l \rceil + 1)T_X, T_X$, and T_X respectively. The critical path of the structure is $T_A + (\lceil log_2^l \rceil + 2)T_X$. The area and time complexities for the proposed multiplier and the other similar multipliers [7], [10], [5], [2], [6] are presented in Table 1 and Table 2, respectively.

Table 1. Comparison of Area Complexities for GF(2^m).

Design	XOR	AND	Register
[7]	$lm + 3l$	lm	$2m + l$
[10]	$lm + (l^2 + l)/2$	lm	$2m + l$
[5]	$lm + (l^2 + l)/2$	lm	$2m + l$
[2]	$69/20m^{log_4^6} - 1/4m^{log_4^2} - 11/5$	$m^{log_4^6}$	$2m - 1$
[6]	$lm + l^2/2 + 3l/2 - 1$	lm	$2m$
[proposed]	$lm + (2l - 1)$	lm	$2m$

Table 2. Comparison of Time Complexities for GF(2^m).

Design	Critical path	Latency (clock cycles)
[7]	$T_A + (\lceil log_2^{2l+1} \rceil) T_X$	$s + 2$
[10]	$T_A + (\lceil log_2^l \rceil + 2) T_X$	s
[5]	$T_A + (\lceil log_2^l \rceil + 2) T_X$	$s + 1$
[2]	$T_A + (1 + 3log_4^m) T_X$	$s + 1$
[6]	$T_A + (\lceil log_2^l \rceil + 2) T_X$	$s + 1$
proposed	$T_A + (\lceil log_2^l \rceil + 2) T_X$	s

The analytical comparisons presented in Table 1 and Table 2 can be better understood by considering a specific field order m and a specific digit size l. By selecting the field to be GF(2^{409}) over an irreducible polynomial $x^{409} + x^{87} + 1$ with a digit-size $l = 8$, the complexities presented in Table 1 and Table 2 are computed and presented in Table 3.

Table 3. Area and time complexities comparison for GF(2^{409}) over $x^{409} + x^{87} + 1$ with $l = 8$.

Design	Area (in terms of number of NAND gate equivalents)	Latency (clock cycles)	Critical Path Delay (ns)	Delay (ns)	Area-Delay Product (NAND equivalents × Delay (ns))
[7]	13780	54	0.27	14.58	200913
[10]	13804	52	0.23	11.96	165096
[5]	13804	53	0.23	12.19	168271
[2]	23299	53	0.59	31.27	728560
[6]	13788	53	0.23	12.19	168076
[proposed]	13732	52	0.23	11.96	164235

We have 65 nm standard library statistics to estimate the time and area requirements. With this technology, the NAND gate equivalents for XOR gate, AND gate, and register are assumed to be 2, 1.25, and 3.75 [1]. The delays for

XOR gate and AND gate are assumed to be 0.04 and 0.03 [1]. It is observed from Table 3, the proposed multiplier requires marginally less hardware when compared with other similar multipliers. It is also observed that the proposed multiplier achieves low area-delay product as well. Hence, the proposed digit-serial sequential multiplier is suitable for IoT edge-devices which typically has a bus width of 8/16/32 bits.

4 Conclusions

In this paper, a new formulation for the digit-serial finite field multiplication is developed. Based on the formulations, an area-efficient digit-serial finite field multiplier over trinomials is presented. It is observed that the proposed multiplier also achieves a reduction in area-delay product as well. This area-efficient sequential digit-level multiplier is suitable for constrained devices such as IoT devices.

References

1. El-Razouk, H., Reyhani-Masoleh, A.: New bit-level serial GF(2^m) multiplication using polynomial basis. In: 2015 IEEE 22nd Symposium on Computer Arithmetic. pp. 129–136. IEEE (2015)
2. Lee, C.Y., Yang, C.S., Meher, B.K., Meher, P.K., Pan, J.S.: Low-complexity digit-serial and scalable SPB/GPB multipliers over large binary extension fields using (b, 2)-way Karatsuba decomposition. IEEE Transactions on Circuits and Systems I: Regular Papers **61**(11), 3115–3124 (2014)
3. Li, S., Da Xu, L., Zhao, S.: The internet of things: a survey. Information Systems Frontiers **17**(2), 243–259 (2015)
4. Lim, C.H., Hwang, H.S.: Fast implementation of elliptic curve arithmetic in GF(p^n). In: International Workshop on Public Key Cryptography. pp. 405–421. Springer (2000)
5. Meher, P.: High-throughput hardware-efficient digit-serial architecture for field multiplication over GF(2^m). In: 2007 6th International Conference on Information, Communications & Signal Processing. pp. 1–5. IEEE (2007)
6. Namin, S.H., Wu, H., Ahmadi, M.: Low-power design for a digit-serial polynomial basis finite field multiplier using factoring technique. IEEE Transactions on Very Large Scale Integration (VLSI) Systems **25**(2), 441–449 (2016)
7. Song, L., Parhi, K.K.: Low-energy digit-serial/parallel finite field multipliers. Journal of VLSI signal processing systems for signal, image and video technology **19**(2), 149–166 (1998)
8. Suárez-Albela, M., Fraga-Lamas, P., Fernández-Caramés, T.: A practical evaluation on RSA and ECC-based cipher suites for IoT high-security energy-efficient fog and mist computing devices. Sensors **18**(11), 3868 (2018)
9. Sunar, B., Koc, C.K.: Mastrovito multiplier for all trinomials. IEEE Transactions on Computers **48**(5), 522–527 (1999)
10. Tang, W., Wu, H., Ahmadi, M.: VLSI implementation of bit-parallel word-serial multiplier in GF(2^{233}). In: The 3rd International IEEE-NEWCAS Conference, 2005. pp. 399–402. IEEE (2005)