Systolic Gaussian Normal Basis Multiplier Architectures Suitable for High-Performance Applications

Reza Azarderakhsh, Mehran Mozaffari Kermani, Siavash Bayat-Sarmadi, and Chiou-Yng Lee

***Abstract*— Normal basis multiplication in finite fields is vastly utilized in different applications, including error control coding and the like due to its advantageous characteristics and the fact that squaring of elements can be obtained without hardware complexity. In this brief, we present decomposition algorithms to develop novel systolic structures for digit-level Gaussian normal basis multiplication over GF(2^m). The proposed architectures are suitable for high-performance applications, which require fast computations in finite fields with high throughputs. We also present the results of our application-specific integrated circuit synthesis using a 65-nm standard-cell library to benchmark the effectiveness of the proposed systolic architectures. The presented architectures for multiplication can result in more efficient and high-performance VLSI systems.**

***Index Terms*— Cryptography, Gaussian normal basis (GNB), security, systolic architecture.**

I. Introduction

Efficient, high-performance, and low-complexity designs of finite field arithmetic and their applications have received much attention in recent years. For instance, many research works have focused on low-complexity and high-performance designs of the arithmetic units for cryptosystems [1]–[8], including Gaussian normal basis (GNB), a special case of normal basis representation of GF(2^m) [9]–[13].

For large finite fields in GF(2^m) (referred to as binary extension fields), field multiplications can be performed by investing on the systolic array approach to achieve high-performance and regular VLSI implementations [14]. Systolic array architectures do not suffer from unappealing irregular circuit designs. Although systolic architectures are well-known to be used in applications needing high-speed structures, they are commonly used when their area complexity is acceptable for special classes of GF(2^m). For example, in [15], an optimal normal basis systolic multiplier is proposed, which is highly regular and can be implemented in digit-serial manner. This systolic multiplier has been employed in [16] for high-performance implementations of reconfigurable hardware to break ECC2K-131, a Koblitz curve challenge over GF(2^{131}).

In [14] and [17], efficient systolic multipliers over GF(2^m) have been presented. Furthermore, efficient digit-serial/digit-level (DL) (or systolic) multipliers are proposed in [18]–[20]. The GNB multiplication is widely implemented through DL parallel-in parallel-out (DL-PIPO) architectures [20] to achieve low latencies in their hardware implementations; however, their critical path delays are relatively high, leading to low-frequency architectures. In order to achieve low latencies and low total computation times and thus high-speed architectures, in this brief, we propose decomposition algorithms to develop systolic GNB multiplier architectures over GF(2^m). The proposed multipliers can achieve low-latency structures as compared with the existing GNB systolic/sequential multipliers [19], [21]. In particular, the presented systolic multipliers over GF(2^m) only require the latency of $\leq 2\lceil\sqrt{m/d}\rceil$ clock cycles, where d is the digit-size. We also compare the hardware and time complexities of the proposed architectures with the previously presented ones through application-specific integrated circuit (ASIC) synthesis to benchmark the higher efficiencies of the presented architectures.

Manuscript received September 18, 2013; revised April 28, 2014 and June 28, 2014; accepted August 3, 2014. Date of publication September 4, 2014; date of current version August 21, 2015.

R. Azarderakhsh is with the Department of Computer Engineering, Rochester Institute of Technology, Rochester, NY 14623 USA (e-mail: rxaeec@rit.edu).

M. Mozaffari Kermani is with the Department of Electrical and Microelectronic Engineering, Rochester Institute of Technology, Rochester, NY 14623 USA (e-mail: m.mozaffari@rit.edu).

S. Bayat-Sarmadi is with the Sharif University of Technology, Tehran 14588-89694, Iran (e-mail: sbayat@sharif.edu).

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).

Digital Object Identifier 10.1109/TVLSI.2014.2345774

II. Preliminaries

The set $N = \{\beta, \beta^2, \beta^{2^2}, \ldots, \beta^{2^{m-1}}\}$ is called normal basis of GF(2^m) for $\beta \in$ GF(2^m) with β being a normal element of GF(2^m). Let us assume that $m > 1$ and $T > 1$ be two integers such that $p = mT + 1$ be a prime number and $\gcd(mT/k,\ m) = 1$, where k is the multiplication order of 2 modulo p. In addition, assume that α be a primitive $mT + 1$th root of unity in GF(2^{Tm}). Then, for any primitive, the Tth root of unity k in $\mathbb{Z}_p$, $\beta = \sum_{i=0}^{T-1} \alpha^{k^i}$ generates a normal basis of GF(2^m) over GF(2), which is called the GNB of type T [22]. The multiplication over GNB is based on a multiplication matrix $\mathbf{M}_{m\times m}$. For simplicity, it is a common practice to store the columns of 1s of the multiplication matrix $\mathbf{M}$ instead of whole $\mathbf{M}$. Therefore, we only need to store those in rows 1 up to $m-1$ and build a new matrix $\mathbf{R}_{(m-1)\times T}$ [23]. Similar property stands for the $\mathbf{R}$ matrix as one has: $R(m-i,\ j) = R(i,\ j)+i \bmod m$, $1 \leq i \leq m-1/2$, $1 \leq j \leq T$. In the rest of this brief, we use the notation of $\mathbf{R}$ matrix instead of the multiplication matrix $\mathbf{M}$.

Recently, low-complexity DL-PIPO GNB multipliers have been proposed in [23] and [24] (optimized in [20]). The time complexity of the DL-PIPO GNB multiplier is $T_A+(\lceil\log_2 T\rceil+\lceil\log_2(d+1)\rceil)T_X$ and its space complexity is dm AND gates and $\leq d(m-1)/2(T-1)+dm$ XOR gates. The area complexity is further reduced by a common subexpression elimination algorithm proposed in [20] to $n_p+v_p(T/2-1)+dm$ XOR gates, where $n_p \leqslant \min\{(v_pT/2), \binom{m}{2}\}$ and $v_p = d(m-1)/2$.

III. Proposed DL Systolic GNB Multipliers

From the symmetric structure of the matrix $\mathbf{R}_{(m-1)\times T}$ presented in Section II, one can write the following for $S(i, B)$ as $S(m-k, B) = S(k, B) \gg k,\ 1 \leq k \leq m-1/2$ [23]. Thus, instead of using matrix $\mathbf{R}_{(m-1)\times T}$, we need to define the matrix $\mathbf{u}_{m-1/2\times T}$ as $\mathbf{u} = [u_k]_{k=1}^{m-1/2}$, where u_k is the row k of the matrix $\mathbf{u}$. The entries of u_k are T integers in the range of $[0, m-1]$ and can be obtained from $\mathbf{R}_{(m-1)\times T}$ as $s'(k, B) = (B \ll R(2k, 1)) \oplus (B \ll R(2k, 2)) \oplus \cdots \oplus (B \ll R(2k, T))$, $1 \leq k \leq m-1/2$. For a detailed information, one needs to refer to [20] and [23].

Now let $q = \lceil m/d\rceil$, $1 \leq d \leq m$, then, one can write the product C as $C = \sum_{i=0}^{q-1} L^{2^{id}}(X \gg id, B \gg id)$,

TABLE I
COMPARISON OF THE TIME AND SPACE COMPLEXITIES FOR VARIOUS DIGIT-SERIAL MULTIPLIERS OVER GF(2^m)

Multiplier	# XOR	# AND	# FF	Latency	CPD
GNB DL-PIPO [20]	$< \frac{d(m-1)}{2}(T-1)+dm$	dm	$3m$	$\lceil \frac{m}{d} \rceil$	T_{W1}
GNB DLGMp [24], [23]	$\leq \frac{d(m-1)}{2}(T-1)+dm$	dm	$3m$	$\lceil \frac{m}{d} \rceil$	T_{W1}
PB Digit-serial systolic [18]	$dm+2d$	dm	$4m+3d+1$	$2\lceil \frac{m}{d} \rceil$	T_{W2}
GNB Lee-Chiou [19]	$d^2+d+mT+1$	d^2	$3.5d^2+8mT$	$d+\frac{mT}{d}(\frac{mT}{d}+1)$	T_A+T_X
Proposed	$\leq \frac{\lceil \sqrt{\frac{m}{d}} \rceil d(m-1)}{2}(T-1)+(1+\lceil \sqrt{\frac{m}{d}} \rceil d)m$	$\lceil \sqrt{\frac{m}{d}} \rceil dm$	$(1+3\lceil \sqrt{\frac{m}{d}} \rceil)m$	$\leq 2\lceil \sqrt{\frac{m}{d}} \rceil$	T_{W1}

T_{W1}=$T_A+(\lceil \log_2 T \rceil + \lceil \log_2 (d+1) \rceil)T_X$, T_{W2}=$T_A+(\lceil \log_2 T \rceil + \lceil \log_2 d \rceil)T_X$.

Algorithm 1 Proposed Systolic Multiplication Scheme

Inputs: $A = (a_0, a_1, \cdots, a_{m-1})$ and B $(b_0, b_1, \cdots, b_{m-1}) \in GF(2^m)$ for m odd.
Output: $C = (c_0, c_1, \cdots, c_{m-1}) = AB$.
1. $C = 0$
2. $X = \bar{A} \gg 1$, where $\bar{A} = (a_{m-1}, a_{m-2}, \cdots, a_1, a_0)$
3. for $i = 0$ to $n-1$ do
4. $C_i = 0$
5. $X_{i,0} = X \gg kid + (k-1)d$
6. $B_{i,0} = B \gg kid + (k-1)d$
7. for $j = 1$ to k do
8. $C_i = (C_i \gg d) + L(X_{i,j-1}, B_{i,j-1})$
9. $X_{i,j} = X_{i,j-1} \ll d$ and $B_{i,j} = B_{i,j-1} \ll d$
10. endfor
11. $C = (C \gg kd) + C_i$
12. endfor

where $L(X, B) = \sum_{j=0}^{d-1} J^{2^j}(X \gg j, B \gg j)$. Assume that n and k are two integers to satisfy $q = kn$. Note that if q is not divisible by k, one needs to append zeros to the least significant bit ends of X and B to satisfy $q = kn$. Let the partial product C_i be defined by $C_i = \sum_{j=0}^{k-1} L^{2^{jd}}(X_i \gg jd, B_i \gg jd)$. According to the above definition, X_i and B_i use integer k, index i, and digit-size d to define its operations such that $X_i = X \gg kid$ and $B_i = B \gg kid$. The product C can be decomposed into n-term partial products as $C = C_0 + C_1^{2^{kd}} + \cdots + C_{n-1}^{2^{(n-1)kd}} = (((C_{n-1})^{2^{kd}} + C_{n-2})^{2^{kd}} + \cdots)^{2^{kd}} + C_0$, where the product C is presented by its most significant digit first (MSD-first). For computing the partial product C_i using the MSD-first scheme, assume that $\overline{X}_i = X_i \gg (k-1)d = X \gg kid + (k-1)d$ and $\overline{B}_i = B_i \gg (k-1)d = B \gg kid + (k-1)d$ are previously determined. Each partial product C_i can then be represented as $C_i = (((L(\overline{X}_i, \overline{B}_i))^{2^d} + L(\overline{X}_i \ll d, \overline{B}_i \ll d))^{2^d} + \cdots)^{2^d} + L(\overline{X}_i \ll (k-1)d, \overline{B}_i \ll (k-1)d)$.

Algorithm 1 describes the proposed systolic GNB multiplication. According to Algorithm 1, Fig. 1 depicts the proposed DL systolic GNB multiplier over GF(2^m). Fig. 1(a) shows the proposed DL systolic multiplier. As one can see, the proposed architecture is composed of k processing elements (PEs) as $\text{PE}_0, \text{PE}_1, \ldots, \text{PE}_{k-1}$ (we use the core operation of the DL GNB multiplier as the PE circuits) and one accumulation circuit (AC). Each PE is carried out by the computations in Steps 8 and 9 of Algorithm 1 and the AC circuit is computed by Step 11. This would reduce the latency of the architectures, which is the main advantage of the proposed structures.

Let us explain the multiplication process presented in Fig. 1(a). Considering the PE operations in Fig. 1(b) and according to Algorithm 1, the output B of PE_j for computing the partial product C_i is denoted by $B_{i,j}$. In the first clock cycle, two elements X_{n-1} and B_{n-1} are fed from left as the inputs to the proposed systolic multiplier for computing the partial result C_{n-1}. In the next clock cycle, two elements X_{n-2} and B_{n-2} are used as the inputs into the proposed systolic multiplier for computing the partial result C_{n-2}, and this is continuously pursued. The GNB multiplication $C = AB$ needs $k + n$ cycles.

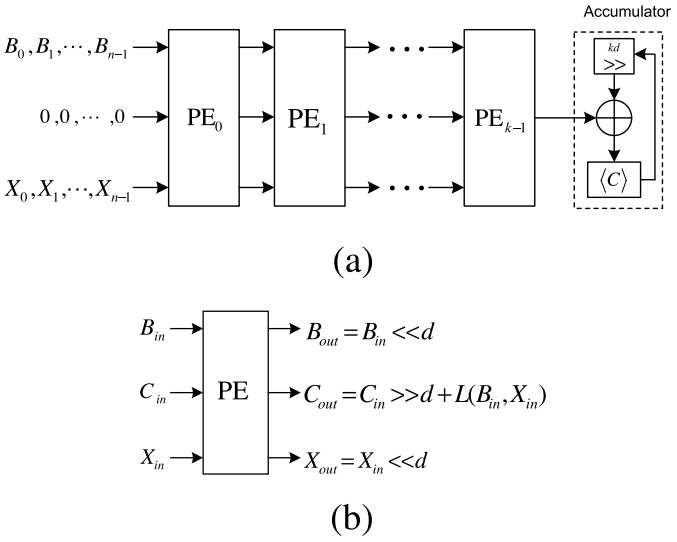

Fig. 1. (a) Proposed DL systolic GNB multiplier over GF(2^m). (b) Detailed PE architecture.

TABLE II
LATENCY COMPARISON OVER GF(2^{409})

Digit-size (d)	2	4	6	8	10	12	14
Proposed	30	22	18	16	14	12	12
DL-PIPO [20]	205	103	69	52	41	35	30
Digit-serial systolic [18]	410	206	138	104	82	70	60

TABLE III
ASIC SYNTHESIS RESULTS FOR THE PROPOSED MULTIPLIER

m, T	Area [μm^2]	CPD [ns]	d	Latency	Time [ns]
163, 4	14,627	1.14	9	10	11.4
	16,439	1.25	11	8	10.0
	39,091	2.53	28	6	15.2
283, 6	110,810	1.61	16	10	16.1
	172,109	1.87	21	8	14.9
	311,186	3.24	36	6	19.4
409, 4	207,222	1.38	13	12	16.6
	264,327	1.63	19	10	16.3
	621,490	3.34	41	8	26.7

Given the proposed DL systolic architecture in Fig. 1, consider having k PEs and one AC, and also note that $q = kn$. Thus, the GNB multiplication can be partitioned into n-term partial results ($C = C_0 + C_1^{2^{kd}} + \cdots + C_{n-1}^{2^{(n-1)kd}}$), where $C_i = \sum_{j=0}^{k-1} L^{2^j}(X_i \gg jd, B_i \gg jd)$. The entire GNB multiplication is computed in $k + n$ clock cycles. We need to minimize $k + n = k + (q/k)$ to have the minimum latency and this requires the first derivative to be equal to zero,

TABLE IV
ASIC (65-nm CMOS LIBRARY) SYNTHESIS RESULTS FOR THE PREVIOUS AND THE PRESENTED MULTIPLIERS OVER GF(2^{409})

Multiplier	Digit-size	Latency	Area [μm^2]	CPD [ns]	Total time [ns]	Total time× area [pm^2s]
Proposed systolic architecture	7	16	128,201	1.28	20.4	2,615
	13	12	207,222	1.38	16.6	3,439
	19	10	232,078	1.63	16.3	3,782
DL-PIPO [23], [24] (optimized in [20])	13	32	71,760	1.44	46.1	3,308
	18	23	115,806	1.55	35.6	4,122
	23	18	147,475	1.68	29.7	4,380
Digit-serial systolic [18]	13	64	58,917	1.23	78.7	4,637

i.e., $1 - q/k^2 = 0$, which yields to $k = \sqrt{q}$. In this regard, $k = n = \lceil \sqrt{q} \rceil$ is selected ($2\lceil \sqrt{m/d} \rceil$ for latency in this case). If m/d is not a perfect square, then, the computation might be performed even in fewer cycles. Thus, for even type-T GNB over GF(2^m), the latency of the proposed DL systolic multiplier is $\leq 2\lceil \sqrt{m/d} \rceil$. The latency of our proposed systolic GNB multiplier is at most $2\lceil \sqrt{m} \rceil$ clock cycles if $d = 1$. In this case, the proposed multiplier needs $\lceil \sqrt{m} \rceil$ PEs.

Using the systolic array implementation in Fig. 1, the presented GNB multiplier includes k PEs and one AC circuit. Each PE circuit is composed of dm AND gates, $\leq d(m-1)/2(T-1) + dm$ XOR gates, and three m-bit registers. The critical path delay of each PE is $T_A + (\lceil \log_2 T \rceil + \lceil \log_2(d+1) \rceil) T_X$. The AC component includes one m-bit GF(2^m) adder and one m-bit register. Given the structure of Fig. 1, the latency of the proposed GNB multiplier is $\leq 2\lceil \sqrt{m/d} \rceil$ clock cycles.

IV. COMPLEXITY BENCHMARK

Table I lists the comparison of some digit-serial multipliers [18]–[20], [23], [24]. According to the table, the proposed digit-serial systolic multipliers have latency of $2\lceil \sqrt{m/d} \rceil$, while a number of existing DL GNB multipliers require the latency of $\lceil m/d \rceil$ clock cycles [20], [23], [24] and the digit-serial systolic Montgomery multiplier using Toeplitz matrix-vector representation requires $2\lceil m/d \rceil$ clock cycles [18]. In addition, the architecture proposed in [19] requires $d + mT/d(mT/d + 1)$ clock cycles, which is more than the ones proposed in this brief. We note that the architecture proposed in [17] is not digit-serial and, hence, we did not include its complexities in this table. We select the field GF(2^{409}) to compare the latencies of various digit-serial multipliers. From Table II, it is shown that the latency of our proposed architectures is lower than those of the existing multipliers. For instance, the proposed multiplier has nearly 1.8–6.8 times less latency compared with those of the existing digit-serial multipliers as digit-size increases from 2 to 14, as seen in Table II.

In terms of area complexity, one should note that the DL-PIPO multiplier [20] requires $< d(m-1)/2(T-1) + dm$ XORs, dm ANDs, and $3m$ flip-flops (FFs), as given in Table I. From Fig. 1, the proposed architecture requires k PEs and one AC component, where $k = \sqrt{m/d}$. The AC component is composed of m-bit FFs and m-bit XOR gates. Hence, the proposed multiplier requires $\leq kd(m-1)/2(T-1) + (1+kd)m$ XORs, kdm ANDs, and $(1+3k)m$ FFs. It is worth mentioning that the architecture proposed in [19] has higher latency and requires more FFs in comparison with our proposed architectures.

V. ASIC SYNTHESIS RESULTS

To investigate the performance of the proposed architecture for digit-serial systolic multipliers, we have implemented them on hardware (ASIC platform) as seen in Table III (with different performance and implementation metrics tabulated and for the frequency of 10 MHz for power derivations). We have used Taiwan Semiconductor Manufacturing Company 65-nm standard-cell library for the ASIC results using Synopsys Design Compiler. Benchmark through hardware platforms is essential in determining the effectiveness of the devised approaches. The timing constraints have been uniformly chosen. These also include all other ASIC constraints to have the same restrictions for all the implemented works. Having same constraints for timing, area, map effort, and the like ensures uniform comparison for all the works, which would also have the same potential area bloats uniformly for all architectures.

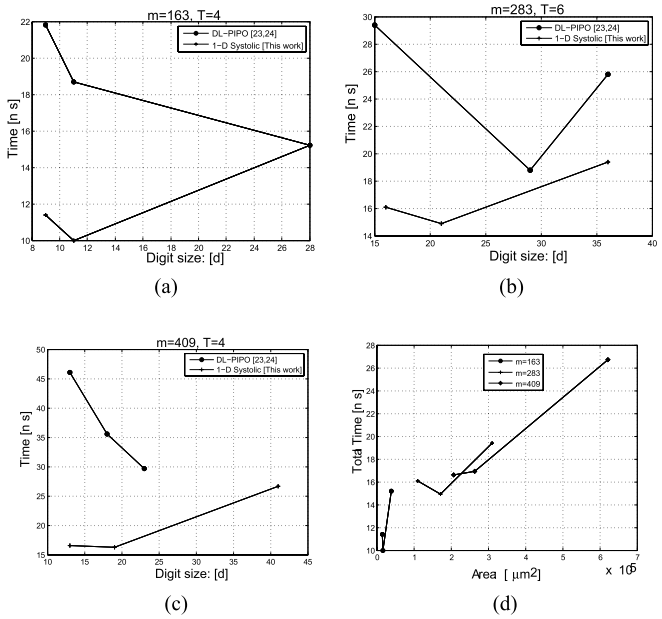

Fig. 2. Comparison of timing results in terms of digit size d for (a) $m = 163$, (b) $m = 283$, and (c) $m = 409$ field sizes for the proposed architecture and the one proposed in [23] and [24]. (d) Area-time ASIC implementation results.

The implementation results for our proposed 1-D systolic architecture are reported in Table III in terms of different digit sizes and field sizes, including $m = 163$, $m = 283$, and $m = 409$. As one can see, our proposed architecture performs better using smaller digit sizes. For the comparison purposes with the previous work, we have synthesized the corresponding existing multipliers and obtained the performance and the synthesis metrics, as shown in Table IV. We have compared our 1-D systolic multiplier with the traditional DL multipliers in terms of different digit sizes for $m = 409$. As seen in this table, at the expense of higher area, the proposed architecture achieves better total time of computation compared with the other structures. We note that for the architecture of [23] and [24], as the digit size increases, the maximum operating clock frequency decreases and the reduction in the latency is not significant. However, for our architecture, smaller digit sizes

(and consequently smaller area) result in much smaller latency and, hence, faster time of computation. These result in higher performance for the proposed systolic structures using the GNB.

For instance, for the digit size of $d = 7$ and $m = 409$, our proposed systolic architecture requires 20.4 ns and occupies only 128 201 μm^2, whereas the fastest results obtained in [23] and [24] requires 29.7 ns and occupies 147 475 μm^2. Therefore, our proposed architecture needs to be employed with smaller digit sizes to save the silicon area usage and achieve faster computation results. In comparison to the systolic multiplier proposed in [18], our architecture performs faster occupying almost similar silicon area. For $m = 13$, our architecture performs 40% faster than the one proposed in [18], occupying only 9% larger silicon area.

In Fig. 2, we compare the timing results for the multiplication over our proposed architecture and the one presented in [23] and [24] for different digit sizes and field sizes. As one can see, our 1-D systolic multiplier performs faster through employing smaller digit sizes which is suitable for high-performance applications. In addition, in this figure, the efficiency results of our proposed multiplier are investigated.

VI. Conclusion

In this brief, GNB multiplication schemes to realize DL systolic GNB multipliers have been presented. The proposed GNB multiplication algorithms are based on decomposition methods leading to new digit-serial systolic architectures. From our theoretical complexity analysis, it is shown that the latency complexities of the proposed multipliers are reduced to $O(\sqrt{m/d})$ clock cycles. Based on our experiments, the proposed architectures are faster and have better performance compared with the existing digit-serial and systolic multipliers available in the literature.

References

[1] H. Fan and M. Hasan, "Subquadratic computational complexity schemes for extended binary field multiplication using optimal normal bases," *IEEE Trans. Comput.*, vol. 56, no. 10, pp. 1435–1437, Oct. 2007.

[2] R. R. Farashahi and M. Joye, "Efficient arithmetic on Hessian curves," in *Proc. Int. Conf. Pract. Theory Public Key Cryptograph.*, May 2010, pp. 243–260.

[3] A. Hariri and A. Reyhani-Masoleh, "Bit-serial and bit-parallel Montgomery multiplication and squaring over GF(2^m)," *IEEE Trans. Comput.*, vol. 58, no. 10, pp. 1332–1345, Oct. 2009.

[4] J. L. Imana, R. Hermida, and F. Tirado, "Low complexity bit-parallel multipliers based on a class of irreducible pentanomials," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 14, no. 12, pp. 1388–1393, Dec. 2006.

[5] J. Adikari, V. S. Dimitrov, and L. Imbert, "Hybrid binary-ternary number system for elliptic curve cryptosystems," *IEEE Trans. Comput.*, vol. 60, no. 2, pp. 254–265, Feb. 2011.

[6] J. Adikari, V. S. Dimitrov, and K. U. Järvinen, "A fast hardware architecture for integer to τNAF conversion for Koblitz curves," *IEEE Trans. Comput.*, vol. 61, no. 5, pp. 732–737, May 2012.

[7] M. Cenk, C. Nègre, and M. A. Hasan, "Improved three-way split formulas for binary polynomial and Toeplitz matrix vector products," *IEEE Trans. Comput.*, vol. 62, no. 7, pp. 1345–1361, Jul. 2013.

[8] M. Cenk, C. Nègre, and M. A. Hasan, "Improved three-way split formulas for binary polynomial multiplication," in *Proc. 18th Int. Workshop Sel. Areas Cryptograph.*, 2011, pp. 384–398.

[9] J. Adikari, V. S. Dimitrov, and R. J. Cintra, "A new algorithm for double scalar multiplication over Koblitz curves," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2011, pp. 709–712.

[10] K. Järvinen and J. Skyttä, "On parallelization of high-speed processors for elliptic curve cryptography," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 16, no. 9, pp. 1162–1175, Sep. 2008.

[11] K. Järvinen and J. Skyttä, "Fast point multiplication on Koblitz curves: Parallelization method and implementations," *Microprocessors Microsyst.*, vol. 33, no. 2, pp. 106–116, Mar. 2009.

[12] V. S. Dimitrov, K. U. Järvinen, M. J. Jacobson, W. Chan, and Z. Huang, "Provably sublinear point multiplication on Koblitz curves and its hardware implementation," *IEEE Trans. Comput.*, vol. 57, no. 11, pp. 1469–1481, Nov. 2008.

[13] R. Azarderakhsh and A. Reyhani-Masoleh, "Efficient FPGA implementation of point multiplication on binary Edwards and generalized Hessian curves using Gaussian normal basis," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 8, pp. 1453–1466, Aug. 2012.

[14] P. K. Meher, "Systolic and non-systolic scalable modular designs of finite field multipliers for Reed-Solomon codec," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 17, no. 6, pp. 747–757, Jun. 2009.

[15] S. Kwon, "A low complexity and a low latency bit parallel systolic multiplier over GF(2^m) using an optimal normal basis of type II," in *Proc. 16th IEEE Symp. Comput. Arithmetic*, Jun. 2003, pp. 196–202.

[16] J. Fan, D. V. Bailey, L. Batina, T. Guneysu, C. Paar, and I. Verbauwhede, "Breaking elliptic curves cryptosystems using reconfigurable hardware," in *Proc. Int. Conf. Field Program. Logic Appl.*, Aug./Sep. 2010, pp. 133–138.

[17] Z. Wang and S. Fan, "Efficient montgomery-based semi-systolic multiplier for even-type GNB of GF(2^m)," *IEEE Trans. Comput.*, vol. 61, no. 3, pp. 415–419, Mar. 2012.

[18] S. Talapatra, H. Rahaman, and J. Mathew, "Low complexity digit serial systolic montgomery multipliers for special class of GF(2^m)," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 18, no. 5, pp. 847–852, May 2010.

[19] C.-Y. Lee and C. W. Chiou, "Scalable Gaussian normal basis multipliers over GF(2^m) using Hankel matrix-vector representation," *J. Signal Process. Syst.*, vol. 69, no. 2, pp. 197–211, Nov. 2012.

[20] R. Azarderakhsh and A. Reyhani-Masoleh, "A modified low complexity digit-level Gaussian normal basis multiplier," in *Proc. 3rd Int. Workshop Arithmetic Finite Fields*, Jun. 2010, pp. 25–40.

[21] R. Azarderakhsh and A. Reyhani-Masoleh, "Low-Complexity Multiplier Architectures for Single and Hybrid-Double Multiplications in Gaussian Normal Bases," *IEEE Trans. Comput.*, vol. 62, no. 4, pp. 744–757, Apr. 2013.

[22] *Digital Signature Standard*, National Institute of Standards and Technology, Gaithersburg, MD, USA, Jan. 2000.

[23] A. Reyhani-Masoleh, "Efficient algorithms and architectures for field multiplication using Gaussian normal bases," *IEEE Trans. Comput.*, vol. 55, no. 1, pp. 34–47, Jan. 2006.

[24] S. Kwon, K. Gaj, C. H. Kim, and C. P. Hong, "Efficient linear array for multiplication in $GF(2^m)$ using a normal basis for elliptic curve cryptography," in *Proc. 6th Int. Workshop Cryptograph. Hardw. Embedded Syst.*, Aug. 2014, pp. 76–91.