ELectronics EXpress

LETTER

IEICE Electronics Express, Vol.13, No.9, 1–6

Very efficient point multiplication on Koblitz curves

Turki F. Al-Somani[a)]

Umm Al-Qura University, Computer Engineering Department,
P.O. Box: 715 Makkah 21955, Saudi Arabia

a) *tfsomani@uqu.edu.sa*

Abstract: This paper presents a very efficient scheme for point multiplication on Koblitz curves. The proposed scheme reduces the critical path in the data dependency graph of the point addition operation and increases the utilization of the three parallel multipliers that are used. The proposed scheme exploits an idle multiplier to perform an extra field operation that will be needed in the next iteration. Furthermore, the proposed scheme is implemented on an Altera Stratix II EP2S180F1020C3 FPGA over $GF(2^{163})$ and compared with the related parallel schemes in the literature. The results show that the proposed scheme outperforms the previous schemes in terms of the area-time (AT) and AT^2 products. Accordingly, the proposed scheme is very attractive for use in high-performance applications.

Keywords: cryptoprocessor, elliptic curve cryptography (ECC), field-programmable gate array (FPGA), Koblitz curves, parallel processing, point multiplication

Classification: Electron devices, circuits, and systems

References

[1] A. J. Menezes, I. F. Blake, S. Gao, R. C. Mullin, S. A. Vanstone and T. Yaghoobian: *Applications of Finite Fields* (Kluwer, Boston, MA, 1993).

[2] D. R. Hankerson, S. A. Vanstone and A. J. Menezes: *Guide to Elliptic Curve Cryptography* (Springer-Verlag, New York, 2004).

[3] IEEE P1363, IEEE Standard Specifications for Public-Key Cryptography (2000).

[4] ANSI X9.62 - 1998, Public Key Cryptography for the Financial Services Industry: Curve Digital Signature Algorithm (ECDSA) (1998).

[5] National Institute of Standards and Technology, Recommended Elliptic Curves for Federal Government Use, Appendix to FIPS 186-2 (2000).

[6] Standards for Efficient Cryptography Group-Certicom Research, SEC 1: Elliptic Curve Cryptography, Version 1.0 (2000) http://www.secg.org/.

[7] Wireless Application Protocol (WAP) Forum, Wireless Transport Layer Security (WTLS) Specification. http://www.wapforum.org/.

[8] K. Järvinen and J. Skyttä: IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **16** (2008) 1162. DOI:10.1109/TVLSI.2008.2000728

[9] R. Azarderakhsh and A. Reyhani-Masoleh: IEEE Trans. Circuits Syst. II, Exp. Briefs **60** (2013) 41. DOI:10.1109/TCSII.2012.2234916

[10] R. Azarderakhsh and A. Reyhani-Masoleh: IEEE Trans. Very Large Scale

DOI: 10.1587/elex.13.20160044
Received January 19, 2016
Accepted March 8, 2016
Publicized March 25, 2016
Copyedited May 10, 2016

Integr. (VLSI) Syst. **20** (2012) 1453. DOI:10.1109/TVLSI.2011.2158595
[11] O. Ahmadi, D. Hankerson and F. Rodríguez-Henríquez: J. Univ. Comput. Sci. **14** (2008) 481. DOI:10.3217/jucs-014-03-0481
[12] K. Järvinen: Integration **44** (2011) 270. DOI:10.1016/j.vlsi.2010.08.001
[13] K. Järvinen and J. Skyttä: Microprocess. Microsyst. **33** (2009) 106. DOI:10.1016/j.micpro.2008.08.002
[14] B. Ansari and M. Anwar Hasan: IEEE Trans. Comput. **57** (2008) 1443. DOI:10.1109/TC.2008.133
[15] J. Adikari, V. S. Dimitrov and R. J. Cintra: Proc. IEEE ISCAS (2011) 709. DOI:10.1109/ISCAS.2011.5937664
[16] D. W. Ash, I. F. Blake and S. A. Vanstone: Discrete Appl. Math. **25** (1989) 191. DOI:10.1016/0166-218X(89)90001-2
[17] R. Azarderakhsh and A. Reyhani-Masoleh: Proc. 3rd Int. WAIFI **6087** (2010) 25. DOI:10.1007/978-3-642-13797-6_3
[18] T. Itoh and S. Tsujii: Inf. Comput. **78** (1988) 171. DOI:10.1016/0890-5401(88)90024-7
[19] C. D. Thompson: Ph.D. dissertation, Carnegie Mellon University, Dep. Comput. Sci. (1980).

1 Introduction

Elliptic Curve Cryptography (ECC) is considered a serious alternative to many public key encryption algorithms. ECC with a key size of 128–256 bits offers security equal to that of RSA with a key size of 1,000–2,000 bits [1, 2]. To date, no significant weaknesses have been identified in the ECC algorithm, which is based on the discrete logarithm problem over points on an elliptic curve. The difficulty of the problem allows the ECC key sizes to be reduced considerably. This advantage of ECC has recently gained remarkable recognition and has been incorporated into many standards such as those of the IEEE, ANSI, NIST, SEC and WTLS [3, 4, 5, 6, 7].

Point multiplication is the basic operation of ECC. The point multiplication of a group of points on an elliptic curve is analogous to the exponentiation of a multiplicative group of integers modulo a fixed integer m. The point multiplication operation, denoted as kP, where k is an integer and P is a point on the elliptic curve, represents the addition of k copies of point P. Point multiplication is then calculated according to a series of point doubling and addition operations of the point P that depend on the bit sequence representing the point multiplier k.

Binary Koblitz curves are a special class of generic curves for which point multiplication can be efficiently computed using their special properties [2]. These curves use a Frobenius map and point addition operation only for computing point multiplication. Binary curves have attracted many researchers to reduce point multiplication. These methods include parallelization, by using multiple parallel field multipliers in the finite field computations [8, 9, 10, 11], and by interleaving [12, 13]. Recently, several methods to perform parallel computations for point addition on Koblitz curves have been proposed in [8, 9, 11, 13, 14, 15]. More recently, it was claimed that four is the maximum number of parallel field multipliers required to achieve the highest parallelization in computing point multi-

DOI: 10.1587/elex.13.20160044
Received January 19, 2016
Accepted March 8, 2016
Publicized March 25, 2016
Copyedited May 10, 2016

plication on Koblitz curves [9]. However, the present paper reduces the critical path of the point addition operation to $3M + 11$ using only three field multipliers, where M is the latency (number of clock cycles) for a digit-level Gaussian normal basis (GNB) multiplication operation [16]. This was achieved by using the idle field multiplier in the data dependency graph of the point addition operation to perform an extra field operation that will be needed in the next iteration (see Fig. 1). For the purpose of comparison with previous implementations, the proposed scheme was implemented on an Altera Stratix II EP2S180F1020C3 FPGA over $GF(2^{163})$.

The contributions of this paper are as follows:

- We reduced the critical path of the addition point operation on Koblitz curves.
- We increased the utilization of the three parallel multipliers.
- We modeled the proposed scheme using VHDL and implemented it on an Altera Stratix FPGA over $GF(2^{163})$.
- We compared our results with the related parallel schemes in the literature on Koblitz curves.

2 Proposed scheme

The data dependency graph of the point addition operation on a Koblitz curve using the mixed Lopez-Dahab coordinate system is shown in Fig. 1(a) [8], where the latency of each step is indicated on the left, where the latency for GNB field addition, squaring and multiplication are 1, 1 and M clock cycles [16], respectively.

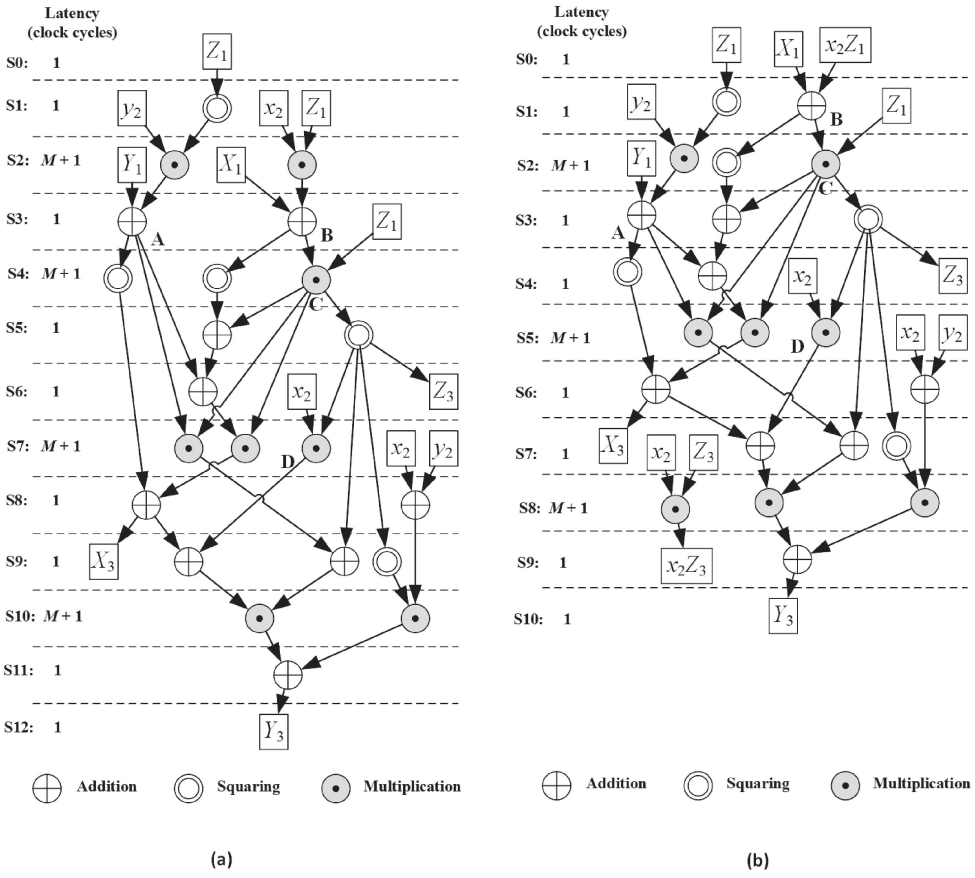

Fig. 1. Data dependency graph of: (a) point addition on a Koblitz curve [8]. (b) the proposed scheme using only three multipliers.

DOI: 10.1587/elex.13.20160044
Received January 19, 2016
Accepted March 8, 2016
Publicized March 25, 2016
Copyedited May 10, 2016

Table I. Point addition cost of related work on Koblitz curves over $GF(2^m)$

Ref.	No. of Multipliers	Latency
[8]	3	$(H(k)-1)\times(4M+13)$
[9]	4	$(H(k)-1)\times(3M+13)$
Ours	3	$(H(k)-1)\times(3M+11)$

As one can see, the latency of point addition is $4M+13$. A key observation is that one of the three multipliers in Step S10 is idle. Accordingly, this multiplier could be used by performing one of the field multiplications of the next iteration, when its operands are ready. After analyzing Fig. 1(a), we found that if we exploit this idle multiplier to compute x_2Z_3 in Step S10, then we do not have to compute x_2Z_3 in Step S2 because Z_3 in iteration i is equal to Z_1 in iteration $i+1$ for $0<i<m-1$. This means that in each point addition, we have to compute the tuple (X_3, Y_3, Z_3, x_2Z_3). Note that for the first iteration, $Z_3=1$ and hence we do not have to compute the field multiplication x_2Z_3. Accordingly, the data dependency graph of the proposed scheme is shown in Fig. 1(b).

The latency of each step in Fig. 1(b) is indicated on the left as in Fig. 1(a). Clearly, Fig. 1(b) shows that the cost of point addition has improved to $3M+11$, which outperforms the work published in the literature thus far (see Table I). Table I shows a comparison of the point addition cost of related work on a Koblitz curve over $GF(2^m)$, where $H(k)$ is the Hamming weight of τ-NAF expansion of k. Furthermore, the use of the parallel multipliers has been increased from 67% to 89%, which makes the proposed scheme very efficient and hence attractive.

3 Implementation and results

To compare the proposed scheme with the previous related work, we modeled the proposed scheme using VHDL and implemented it on an Altera Stratix II EP2S180F1020C3 FPGA. Fig. 2 shows the architecture of the implemented cryptoprocessor, which consists of the controller, register file and the field arithmetic unit (FAU). The controller is designed as a finite state machines (FSM) to perform point multiplication. The register file holds the values of the base point

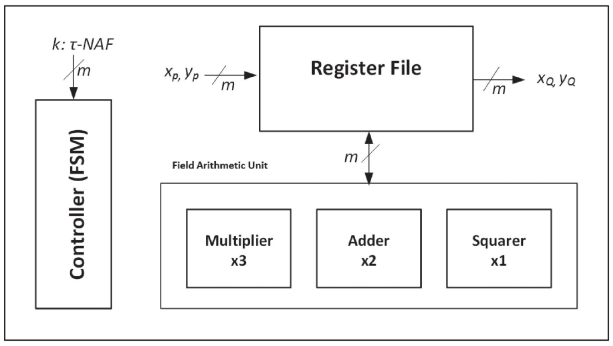

Fig. 2. The proposed architecture for point multiplication on Koblitz curves.

DOI: 10.1587/elex.13.20160044
Received January 19, 2016
Accepted March 8, 2016
Publicized March 25, 2016
Copyedited May 10, 2016

$P = (x_p, y_p)$, the intermediate values and the final results. The FAU, however, contains three multipliers, two adders and one squarer. Once k is available in the τ-NAF representation, at the input of the controller, the FAU starts the computations.

We implemented the presented multiplier in [9], which is based on [10, 17], over $GF(2^{163})$ with two digit sizes = 33 and 41, for fair comparison. In a digit-level parallel-in parallel-out GNB multiplier, the results are available in parallel after $q = \lceil m/d \rceil$ clock cycles, where d is the digit size. Therefore, as the pipelining adds one extra clock cycle to the latency of multiplication, the latency of the multiplier is given by $M = \lceil m/d \rceil + 1$, where $1 \leq d \leq m$. The Itoh-Tsujii inverter [18], however, is used to convert the final results back to affine coordinates using the FAU, which requires $11M + 11$. Finally, the cost of performing the Frobenius maps is m clock cycles. Accordingly, the total latency for performing point multiplication over $GF(2^{163})$ without τ-NAF conversion is $(H(k) - 1) \times (3M + 11) + 163 + 11M + 11$, where $(H(k) - 1) \times (3M + 11)$ is the latency of computing the point addition operation.

The lower bound on the area-time cost of a given design is usually used as a performance metric (area) $\times$ (time)$^{2\alpha}$, $0 \leq \alpha \leq 1$, where the choice of α determines the relative importance of area and time. Such lower bounds have been obtained for several problems, e.g., discrete Fourier transform, matrix multiplication, binary addition and others [19]. Once the lower bound on the chosen performance metric is known, designers attempt to devise algorithms and designs that are optimal for a range of area and time values. Even though a design might be optimal for a certain range of area (A) and time (T) values, it is nevertheless of interest to obtain designs for minimum values of time, i.e., for maximum speed performance and for minimum area. Accordingly, both the area-time (AT) and AT^2 products are used here to compare the proposed scheme with the other previous designs.

Table II shows a comparison of the proposed scheme with the related work over $GF(2^{163})$ on the Altera Stratix II EP2S180F1020C3 FPGA. The second and the third columns in this table show the number of used multipliers (M) and the digit size d, respectively. The fourth column shows the total latency (L) of point multiplication. The fifth and the sixth columns show the obtained results from Atera's Quartus II design software, which are the maximum frequency and area in terms of adaptive logic modules (ALMs). The seventh column shows the time in

Table II. Results of related work over $GF(2^{163})$ on the Altera Stratix II EP2S180F1020C3 FPGA

Ref.	M	d	L	f_{max} MHz	Area (ALMs)	Time [μs]	AT	AT^2
[8]	3	33	4248	146.7	22,416	28.95	0.64	18.5
[13]	4	19	1422	164.4	25,366	8.64	0.22	1.9
[13]	4	17	1540	162.4	23,580	9.48	0.22	2.1
[9]	4	41	1721	188.7	23,084	9.15	0.21	1.9
[9]	4	33	1892	192.5	18,964	9.85	0.18	1.7
Ours	3	41	1616	187.9	18,236	8.60	0.16	1.4
Ours	3	33	1787	189.8	15,160	9.41	0.14	1.3

DOI: 10.1587/elex.13.20160044
Received January 19, 2016
Accepted March 8, 2016
Publicized March 25, 2016
Copyedited May 10, 2016

μs, which is the result of multiplying the latency by the maximum frequency. Finally, the last two columns show area-time (AT) and AT^2 products [19].

The results in Table II indicate that the proposed scheme achieved the best latency when the digit-size d was $d = 41$. However, the proposed scheme achieved the best area when $d = 33$. Clearly, the proposed scheme outperforms the other schemes for both the AT and AT^2 results.

4 Conclusion

A very efficient point multiplication scheme on Koblitz curves is presented. The proposed scheme reduces the critical path in the data dependency graph of the point addition operation and increases the utilization of the three parallel multipliers that are used. The proposed scheme exploits an idle multiplier to perform an extra field operation that will be needed in the next iteration. Furthermore, the proposed scheme is implemented on an Altera Stratix II EP2S180F1020C3 FPGA over $GF(2^{163})$ and compared with the related parallel schemes in the literature. The results show that the proposed scheme outperforms the previous schemes in terms of the AT and AT^2 products. The proposed scheme is very efficient and hence attractive for use in high-performance applications.

Acknowledgments

The author would like to acknowledge the support of both Umm Al-Quara University and King Abdulaziz City for Science and Technology.

DOI: 10.1587/elex.13.20160044
Received January 19, 2016
Accepted March 8, 2016
Publicized March 25, 2016
Copyedited May 10, 2016