Journal of Network Intelligence
Taiwan Ubiquitous Information
 ISSN 2414-8105 (Online)
Volume **6**, Number **1**, February **2021**

FPGA Implementation of Binary Edwards Curve Point Addition

Jeng-Shyang Pan

Fujian Provincial Key Laboratory of Big Data Mining and Applications
Fujian University of Technology
Fuzhou 350118, China
College of Computer Science and Engineering
Shandong University of Science and Technology
Qingdao 266590, China
jengshyangpan@gmail.com

Pei-Cheng Song

College of Computer Science and Engineering
Shandong University of Science and Technology
Qingdao 266590, China
spacewe@outlook.com

Qing-Yong Yang*

College of Computer Science and Engineering
Shandong University of Science and Technology
Qingdao 266590, China
*Correponding Author: qyzide@163.com

Lyu-Chao Liao

Fujian Provincial Universities Key Laboratory of Industrial Control and Data Analysis
Fujian University of Technology
Fuzhou, 350118, China
lcliao@csu.edu.cn

Received October 2020; revised January 2021

Abstract. *Currently, information security is becoming more and more important. Elliptic Curve Cryptography (ECC) has attracted more attention due to its high security performance and short key length, and has been more and more used in the process of information encryption. This article provides the corresponding FPGA hardware design scheme for the binary Edwards curve in ECC. At the same time, four schemes are analyzed and compared for the multiplier module in the finite field module. Finally, using the relevant parameters of $GF(2^{163})$ in the NIST standard, the behavior simulation experiment of the Point Addition module of the binary Edwards curve is carried out to prove the feasibility of the scheme.*
Keywords: Elliptic Curve Cryptography; FPGA; NIST; binary Edwards curve.

1. **Introduction.** With the rapid development of Internet of Things (IoT) technology, the scale of the industry continues to expand. More and more IoT smart devices are put into actual production and life, which brings convenience, but also creates some safety problems [1]. Therefore, how to encrypt and protect all kinds of personal sensitive

information and ensure the integrity and secrecy of the information when it is transmitted and stored on the network is particularly important.

Elliptic Curve Cryptography (ECC), as a research category of cryptography, has a rich research history [2–7]. Compared with the RSA cryptosystem, under the same security conditions, ECC has the advantages of higher encryption and decryption efficiency, faster calculation speed, and less storage space and bandwidth [8]. It is suitable for use in environments with limited computing resources and high-speed communications [9–13]. And with the continuous development of Internet of Things technology, the security of Internet of Things devices has always been a problem that needs to be solved. Therefore, the application of ECC on IoT devices has become an effective method to solve this problem [14]. In recent years, there have been more achievements in the research and implementation of elliptic curve encryption algorithms [15–18].

The realization of ECC algorithm is divided into software realization and hardware realization [16]. With the continuous development of Internet of Things technology, the realization method of ECC algorithm gradually turns to the realization of hardware system. The hardware realization is divided into two kinds of realization schemes of Field Programmable Gate Arrays (FPGA) and Application Specific Integrated Circuit (ASIC). ASIC has disadvantages such as poor flexibility, poor scalability, and high development cost when it is implemented. Once the chip is determined, it is extremely difficult to modify the hardware structure, which is not convenient for secondary development. Compared with it, FPGA has higher programmability, less development cost and shorter development cycle. While obtaining higher performance, it has the advantages of convenient upgrade and maintenance. Therefore, it is a better choice to use FPGA to implement ECC algorithm [19, 20]. However, the complex mathematical operations of the ECC algorithm itself and the diversity of applications also pose challenges to FPGA implementation [21]. Therefore, while giving the relevant realization principle of the elliptic curve, this article also gives the corresponding hardware design scheme, and carries on the corresponding simulation test to verify the feasibility of the scheme.

The remaining chapters of this article are arranged as follows: Section 2 introduces the overall scheme of Edwards elliptic curve encryption algorithm. Section 3 introduces several specific algorithms implemented in this article. Section 4 shows the simulation test results of the design scheme. Section 5 concludes this article.

2. **The overall scheme of Edwards elliptic curve encryption algorithm.**

The hardware implementation of the elliptic curve encryption algorithm not only requires the selection of curves and curve parameters, but also the selection and design of algorithms at different levels. The first is the selection and parameter setting of the elliptic curve algorithm. Elliptic curves have many different algebraic equation forms, such as Weierstrass type, Jacobi type, Edwards type, Huff type and Hessian type, etc. Different forms of elliptic curves have different advantages in different applications. At the same time, due to different forms, the specific implementation of the algorithm is also different [14].

Compared with the classic Weierstrass type, the Edwards type has more advantages from the perspective of computational efficiency and security. Therefore, this paper selects the Edwards curve for hardware implementation of related algorithms in an attempt to further improve the efficiency and security in the practical application of elliptic curves.

There are two types of Edwards curves. Because this paper is based on the binary finite field $GF(2^m)$ for implementation, this paper chooses the binary Edwards curve [22, 23].

The binary field $GF(2^m)$ refers to the m-th extended field of the field $GF(2)$=$\{0,1\}$, m is a prime number, in this article $m = 163$, based on the binary Edwards curve

$$E_{B,d_1,d_2} : d_1(x,y) + d_2(x^2,y^2) = xy + xy(x+y) + x^2y^2 \quad (1)$$

In Eq.1, d_1 and d_2 belong to finite field k and d_1 is not equal to 0, and $d_1^2 + d_1 + d_2 \neq 0$.

Based on the actual application requirements of the elliptic curve encryption algorithm, the base point p, the order of the base point, and the private key k need to be determined and stored in advance. For the binary Edwards curve, d_1 and d_2 need to be determined.

Then is the realization of elliptic curve encryption algorithm, which mainly includes two parts: finite field operation and elliptic curve operation [24]. The two parts respectively contain several small components. The finite field operations include four parts: modular addition, modular multiplication, modular squaring and modular inverse. Elliptic curve operations include point doubling and point multiplication. Different parts use FPGA to implement sub-modules. This article realizes the realization of finite field operations on $GF(2^m)$. Finite field operations have two expressions: polynomial basis and normal basis. Since polynomial basis is more suitable for hardware implementation, this paper implements related finite field algorithms based on polynomial basis.

The finite field modular addition operation on $GF(2^m)$ is equivalent to XOR, so the execution efficiency is very high and there is no need to design a separate module. Modular squaring usually has a fixed design method, which will not be introduced in this article. Since the modular inverse operation can be implemented based on modular squaring and modular multiplication, the modular inverse module is no longer designed separately, but implemented by calling modular multiplication and modular squaring modules. As the core module in elliptic curve finite field arithmetic, the realization of modular multiplication will consume a lot of resources [25]. A reasonable and effective algorithm can reduce resource consumption and improve computational efficiency [26, 27]. The most basic binary finite field algorithms include bit serial multipliers, bit parallel multipliers and hybrid multipliers [28–30]. The space complexity of the bit-serial multiplier is small, but it takes a lot of time [31], and the bit-parallel multiplier consumes less time, but the space complexity is quite high [32]. The digital serial multiplier can achieve a compromise between time complexity and space complexity [33]. In recent years, 2-way Karatsuba-Ofman algorithm (2-way KA) and 3-way Karatsuba-Ofman algorithm (3-way KA) are often used to reduce the space complexity of modular multiplication. In addition, there are algorithms such as (a,2)-way KA [34], (a,b)-way KA [35]. This article attempts to implement the 2-way KA and conduct an overall simulation test.

Compared with other elliptic curve forms, the binary Edwards curve is more efficient. Its addition formula is symmetrical and unified. The point addition and doubling formula are the same, and there is no need to design two different modules. After completing the implementation of the finite field operation module, you can choose to use affine coordinates or projective coordinates based on the calculation efficiency of the finite field operation module. The use of projective coordinates can avoid the use of modular inverse operations, and the use of affine coordinates requires inversion operations, but it will reduce the number of modular multiplication operations. In the specific hardware implementation, you can choose according to the overall effect.

Since the point multiplication operation of the binary Edwards curve requires a lot of point operations, it is also possible to improve the calculation efficiency by separately designing algorithms for different forms of point operations, such as the direct calculation of multiple points. Related algorithms can make full use of the intermediate value in the calculation process to improve the calculation efficiency of different forms of point operations, such as conjugate plus point operations. Next, we will specifically introduce

the related algorithms of modular multiplication in the finite field operation of hardware implementation and the point operation of the binary Edwards curve.

3. **The specific principles of several operations.** This section mainly introduces the specific principles of several modular multiplication algorithms and binary Edwards curve operations described in the second section.

The first is the basic principle of the bit-serial multiplier. The bit-serial multiplier is based on the principle of binary finite field multiplication. The basic principle of binary finite field multiplication is shift addition. The specific formula is as follows

$$c(z) = a(z) \cdot b(z) = a_{m-1} z^{m-1} b(z) + \cdots + a_2 z^2 b(z) + a_1 z b(z) + a_0 b(z) \tag{2}$$

Since $m = 163$ in this article, in the process of multiplication, the multiplication of two polynomials with the highest power of 163 requires modular reduction. According to the NIST standard, the reduced polynomial $f(z) = z^{163} + z^7 + z^6 + z^3 + 1$ selected in this article [24]. Then the formula of the bit-serial multiplier is

$$c(z) = (\cdots((a_{m-1}b(z)z + a_{m-2}b(z))z + a_{m-3}b(z))z + \cdots + a_1 b(z))z + a_0 b(z) \bmod f(z) \tag{3}$$

The second is the bit parallel multiplier, the specific calculation formula is

$$\begin{aligned} c(z) &= a(z) * b(z) \bmod (f(z)) \\ &= a_0\ b_0 + (a_0\ b_1 + a_1\ b_0)\, z^1 + \cdots + a_{m-1} b_{m-1} x^{2m-2} \bmod (f(z)) \\ &= a_0\ b_0 \oplus (a_0\ b_1 + a_1\ b_0)\, z^1 \oplus \cdots \oplus a_{m-1} b_{m-1} z^{2m-2} \bmod (f(z)) \\ &= C_0 \oplus C_1 z^1 \oplus \cdots \oplus C_{2m-2} z^{2m-2} \bmod (f(z)) \end{aligned} \tag{4}$$

Based on the above formula, $C_0, C_1, ..., C_{2m-2}$ can be calculated separately in parallel, and the calculation can be completed in one clock cycle, but the space complexity is high and requires a lot of resources. Compared with the bit-parallel multiplier and the bit-serial multiplier, the 2-way KA algorithm is a compromise, and the space complexity and time complexity are more balanced.

Since $a(z)$ represents the polynomial with the highest power of 162, the 2-way KA algorithm divides the entire polynomial into two parts, using A_0 to represent the first half of $a(z)$ and A_1 to represent the second half, then $a(z) = A_0 + z^{m/2} A_1$, similarly, $b(z) = B_0 + z^{m/2} B_1$. Then the multiplication of $a(z)$ and $b(z)$ is as follows

$$\begin{aligned} c(z) = a(z)b(z) &= \left(A_0 + z^{\frac{n}{2}} A_1\right)\left(B_0 + z^{\frac{n}{2}} B_1\right) \\ &= C_0 + C_1 z^{\frac{n}{2}} + C_2 z^n \end{aligned} \tag{5}$$

For Eq.5, $P_0 = A_0 B_0$,$P_1 = (A_0 + A_1)(B_0 + B_1)$, $P_2 = A_1 B_1$.Then, $C_0 = P_0, C_1 = P_0 + P_1 + P_2, C_2 = P_2$. Based on the above principles, this article uses FPGA to quickly implement the algorithm. It should be noted that when using the 2-way KA algorithm, it is necessary to ensure that m is an even number. This article has made some simple modifications in the implementation process.

Next is an introduction to the binary Edwards curve addition formula. Assume that two points $P(x_1, y_1)$ and $\mathrm{Q}(x_2, y_2)$ are added together. In order to calculate the added point (x_3, y_3). The additive formula of the binary Edwards curve additive group is defined as

$$
\begin{aligned}
&(x_3, y_3) = (x_1, y_1) + (x_2, y_2) \\
&x_3 = \frac{d_1 (x_1 + x_2) + d_2 (x_1 + y_1)(x_2 + y_2) + (x_1 + x_1^2)(x_2 (y_1 + y_2 + 1) + y_1 y_2)}{d_1 + (x_1 + x_1^2)(x_2 + y_2)} \\
&y_3 = \frac{d_1 (y_1 + y_2) + d_2 (x_1 + y_1)(x_2 + y_2) + (y_1 + y_1^2)(y_2 (x_1 + x_2 + 1) + x_1 x_2)}{d_1 + (y_1 + y_1^2)(x_2 + y_2)}
\end{aligned} \tag{6}
$$

For Eq.6, $d_1, d_2 \in E_{B,d_1,d_2}$. Eq.6 is uniform. When $P = Q$, this formula can also be used for calculation.

The binary Edwards curve addition formula can be based on the affine coordinate system or the projective coordinate system. This article uses the point addition and doubling point formula in the projective coordinate system to avoid the inversion calculation. Let $P = (X_1, Y_1, Z_1)$, $Q = (X_2, Y_2, Z_2)$, then (X_3, Y_3, Z_3) after adding two points can be obtained by the following formula

$$
\begin{aligned}
&W_1 = X_1 + Y_1; W_2 = X_2 + Y_2; A = X_1 \cdot (X_1 + Z_1); B = Y_1 \cdot (Y_1 + Z_1) \\
&C = Z_1 \cdot Z_2; D = W_2 \cdot Z_2; E = d_1 C^2; H = (d_1 Z_2 + d_2 W_2) \cdot W_1 \cdot C \\
&I = d_1 C \cdot Z_1; U = E + A \cdot D; V = E + B \cdot D; S = U \cdot V \\
&X_3 = S \cdot Y_1 + (H + X_2 \cdot (I + A \cdot (Y_2 + Z_2))) \cdot V \cdot Z_1 \\
&Y_3 = S \cdot X_1 + (H + Y_2 \cdot (I + B \cdot (X_2 + Z_2))) \cdot U \cdot Z_1; Z_3 = S \cdot Z_1
\end{aligned} \tag{7}
$$

Based on the introduction of the above specific principles, this article will implement the above specific principles in FPGA, and evaluate and analyze the overall plan.

4. **Experimental results.** This section is based on FPGA to realize the finite field operation of the binary Edwards curve and some point operation algorithms. The simulation test software used in this article is Vivado 2017, and it is implemented using Verilog. The selected development board is the ZYNQ 7000 series of Xilinx. And the maximum values of LUT, FF and BUFG are 53200, 106400 and 32.

In this paper, the selected m value is 163. According to the $NIST$ standard, the corresponding domain polynomial is $f(z) = z^{163} + z^7 + z^6 + z^3 + 1$. Table 1 and Figure 1-6 record the resource occupancy and simulation waveforms of the finite field. The number of bits processed in each cycle of the bit Serial Parallel hybrid Multiplier (multi_sp) is set to 4.

Table 1. Resource occupancy of finite field operations

	LUT		FF		BUFG	
	Estimation	Utilization %	Estimation	Utilization %	Estimation	Utilization %
multi_s	267	0.5	666	0.63	1	3.13
multi_p	12748	23.96	0	0	0	0
multi_sp	513	0.96	665	0.63	1	3.13
multi_2way	677	1.27	995	0.94	1	3.13
sqr	84	0.16	0	0	0	0
inverse	2782	5.23	826	0.78	1	3.13

The following results can be obtained from the resource consumption and simulation time of the four multipliers. The bit Serial Multiplier (multi_s) takes the least resources, but the simulation time is too long. Although the simulation time of bit full Parallel Multiplier (multi_p) is the shortest, it needs the most resources, so it is not suitable for running on resource limited devices. The bit Serial-Parallel hybrid Multiplier (multi_sp) provides a good compromise. On the basis of the serial multiplier, the operation speed is improved by processing multiple bits of data in parallel in each cycle. The 2-way KA

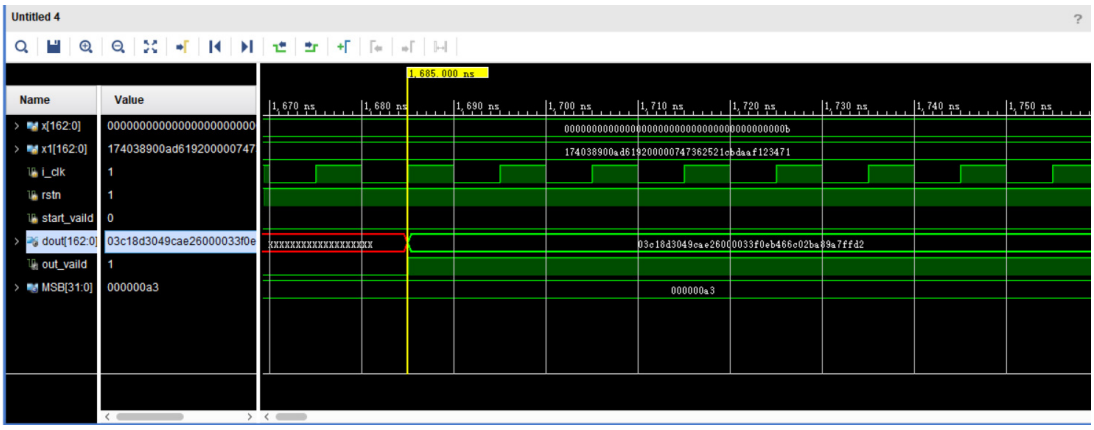

FIGURE 1. Simulation waveform of bit Serial Multiplier (multi_s)

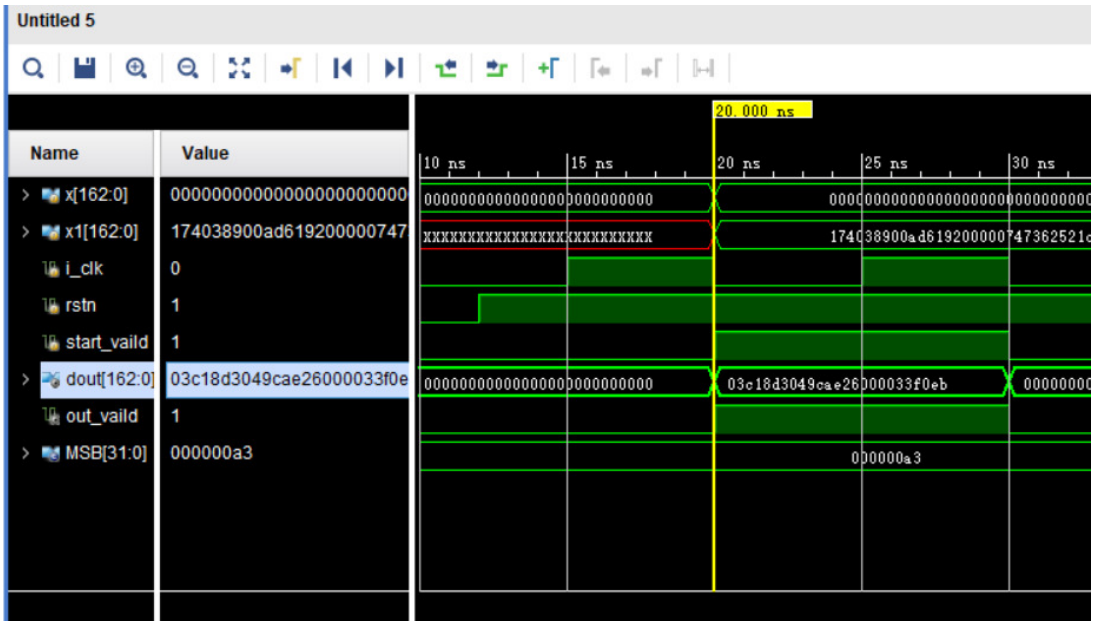

FIGURE 2. Simulation waveform of bit full Parallel Multiplier (multi_p)

Multiplier (multi_2way) is also a compromise solution, which is faster than the bit Serial-Parallel hybrid Multiplier in terms of operating speed. However, due to the process of dividing data, a certain amount of resource consumption is increased. The preprocessing of the data in the Square operation part is achieved by using the generate statement in Verilog. And according to the law of data modulo operation, fully parallel modulo operation is realized. Therefore, the square operation of data can be realized in one clock cycle. The inverse operation is the operation that takes up the most resources in the entire finite field operation part. In this article, the extended Euclidean algorithm is used to achieve the inverse operation.

The next step is to implement the Point Addition algorithm of the binary Edwards curve based on the completed finite field calculation module. Parameters d_0, d_1 and Z are set to 1 respectively. The resource occupation results are shown in Table 2, and the simulation test result of Point Addition operation is shown in Figure 7.

5. **Conclusions.** Aiming at the binary Edwards curve in Elliptic Curve Cryptography (ECC), this paper presents the FPGA hardware design scheme of the Point Addition

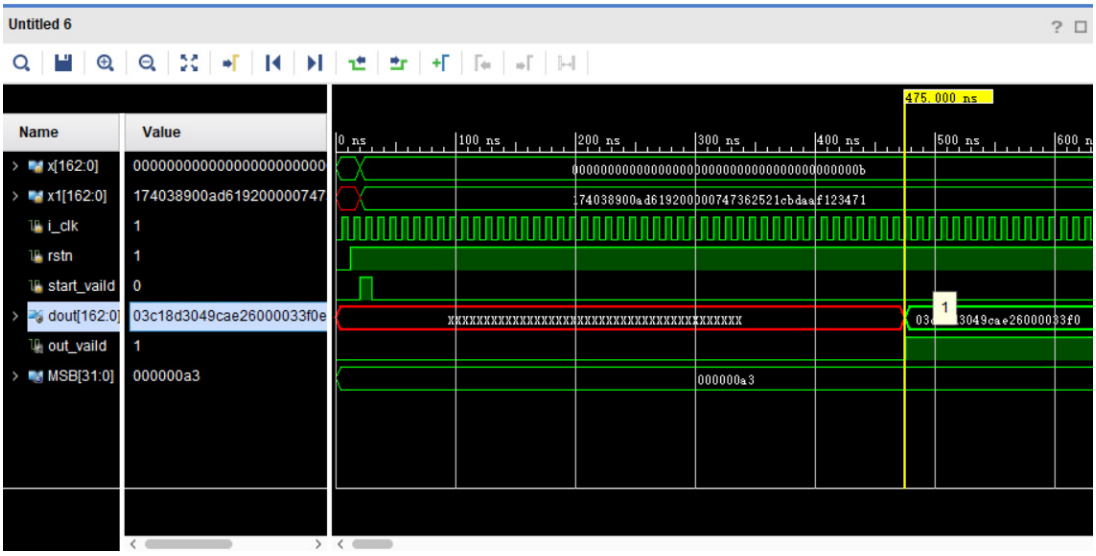

FIGURE 3. Simulation waveform of bit Serial Parallel Hybrid Multiplier (multi_sp)

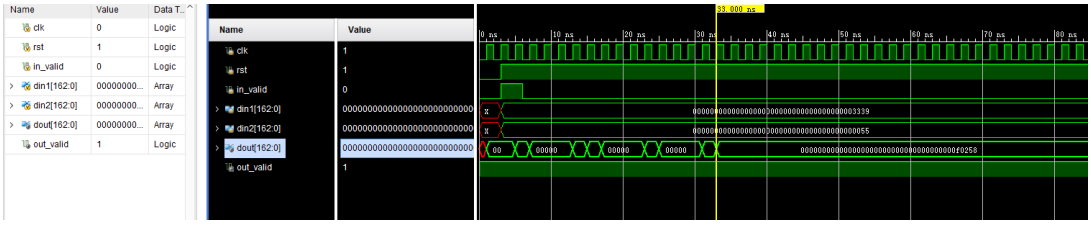

FIGURE 4. Simulation waveform of 2-way KA Multiplier (multi_2way)

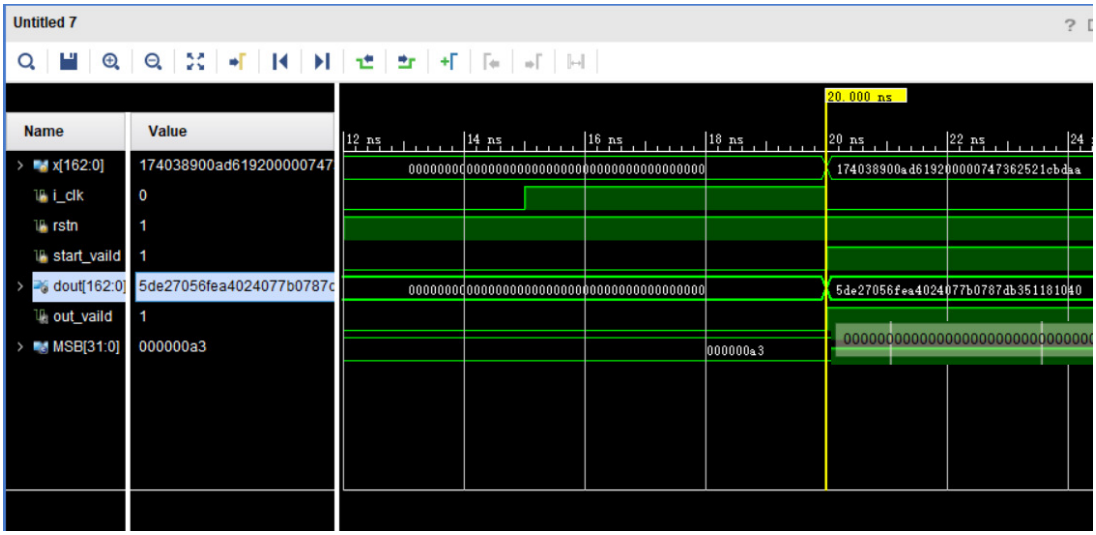

FIGURE 5. Simulation waveform of Square operation (sqr)

TABLE 2. Resource occupancy of Point Addition

	LUT		FF		BUFG	
	Estimation	Utilization %	Estimation	Utilization %	Estimation	Utilization %
Point_add	4711	8.86	5453	5.13	1	3.13

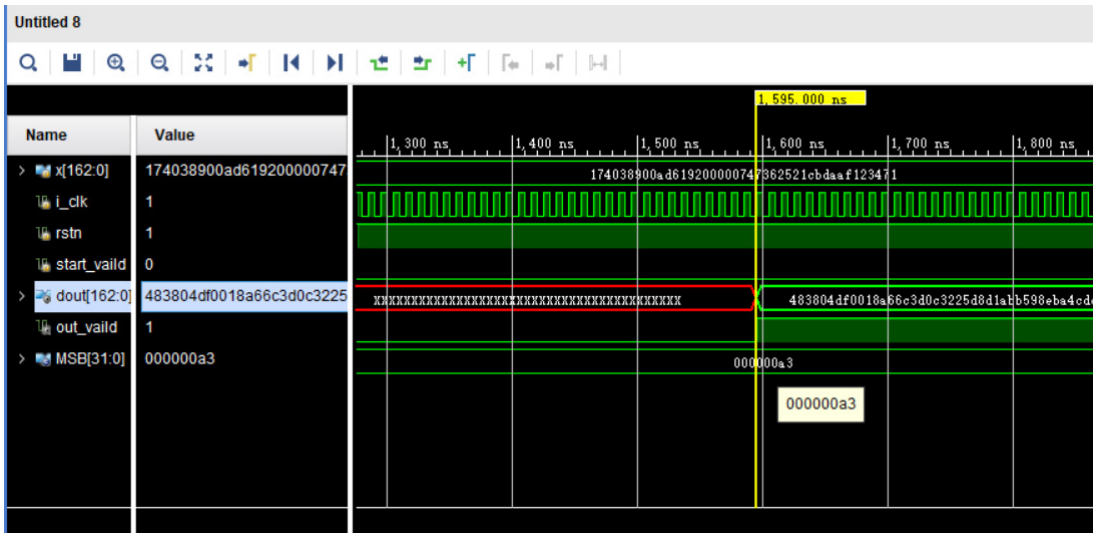

FIGURE 6. Simulation waveform of Inverse operation (inverse)

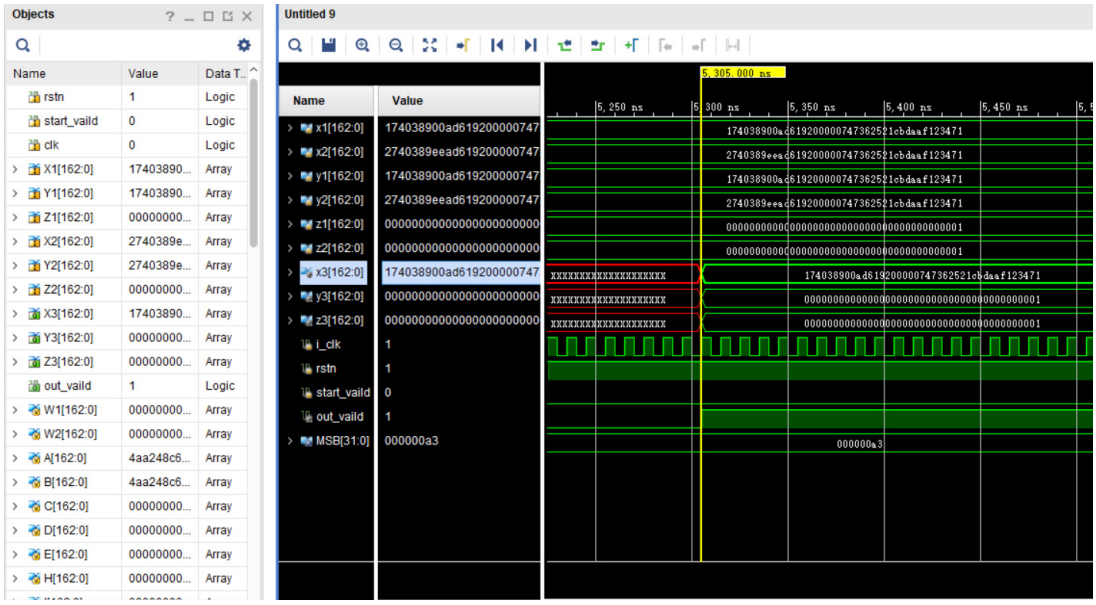

FIGURE 7. Simulation waveform of Point Addition

part. Four kinds of multipliers, square operation and inverse operation in finite field are designed by behavior simulation, and the multipliers are analyzed and compared. Finally, based on the realization of the finite field part, the $GF(2^{163})$ related parameters in the $NIST$ standard are used to realize the FPGA realization and verification of the binary Edwards Point Addition module. In the future, on the basis of this work, we will continue to complete the design and optimization of the binary Edwards Point Multiplication and Point Doubling modules to achieve a complete binary Edwards curve calculation.

Acknowledgment. This work is partially supported by the Natural Science Foundation of Fujian Province (2018J01638). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] L. Atzori, A. Iera, and G. Morabito, The internet of things: A survey, *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[2] D. Hankerson, A. J. Menezes, and S. Vanstone, *Guide to elliptic curve cryptography*. Springer Science & Business Media, 2006.

[3] M. Moussa, E. Badr, and S. Almotairi, A data hiding algorithm based on DNA and elliptic curve cryptosystems, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 10, no. 3, pp. 458–469, 2019.

[4] W. Chang, H. Li, and S. Yin, Mixed symmetric key and elliptic curve encryption scheme used for password authentication and update under unstable network environment, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 8, no. 3, pp. 632–639, 2017.

[5] T.-Y. Wu, T. Wang, Y.-Q. Lee, W. Zheng, S. Kumari, and S. Kumar, Improved authenticated key agreement scheme for fog-driven iot healthcare system, *Security and Communication Networks*, vol. 2021, 6658041, 2021.

[6] T.-Y. Wu, L. Yang, Z. Lee, C.-M. Chen, J.-S. Pan, and S. Islam, Improved ecc-based three-factor multiserver authentication scheme, *Security and Communication Networks*, vol. 2021, 6627956, 2021.

[7] T.-Y. Wu, Y.-Q. Lee, C.-M. Chen, Y. Tian, and N. A. Al-Nabhan, An enhanced pairing-based authentication scheme for smart grid communications, *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–13. DOI: 10.1007/s12652-020-02740-2.

[8] N. Gura, A. Patel, A. Wander, H. Eberle, and S. C. Shantz, Comparing elliptic curve cryptography and RSA on 8-bit CPUs, in *International workshop on cryptographic hardware and embedded systems*, pp. 119–132, 2004.

[9] C.-M. Chen, B. Xiang, K.-H. Wang, Y. Zhang, and T.-Y. Wu, An efficient and secure smart card based authentication scheme, *Journal of Internet Technology*, vol. 20, no. 4, pp. 1113–1123, 2019.

[10] C.-M. Chen, B. Xiang, K.-H. Wang, K.-H. Yeh, and T.-Y. Wu, A robust mutual authentication with a key agreement scheme for session initiation protocol, *Applied Sciences*, vol. 8, no. 10, 1789, 2018.

[11] K.-H. Wang, C.-M. Chen, W. Fang, and T.-Y. Wu, A secure authentication scheme for internet of things, *Pervasive and Mobile Computing*, vol. 42, pp. 15–26, 2017.

[12] C.-T. Li, T.-Y. Wu, C.-L. Chen, C.-C. Lee, and C.-M. Chen, An efficient user authentication and user anonymity scheme with provably security for iot-based medical care system, *Sensors*, vol. 17, no. 7, 1482, 2017.

[13] T.-Y. Wu, C.-M. Chen, K.-H. Wang, J.-S. Pan, W. Zheng, S.-C. Chu, and J. F. Roddick, Security analysis of rhee et al.'s public encryption with keyword search schemes: A review, *Journal of Network Intelligence*, vol. 3, no. 1, pp. 16–25, 2018.

[14] M. Amara and A. Siad, Elliptic curve cryptography and its applications, in *International workshop on systems, signal processing and their applications, WOSSPA*, pp. 247–250, 2011.

[15] J. W. Bos, J. A. Halderman, N. Heninger, J. Moore, M. Naehrig, and E. Wustrow, Elliptic curve cryptography in practice, in *International Conference on Financial Cryptography and Data Security*, pp. 157–175, 2014.

[16] D. Hankerson, J. L. Hernandez, and A. Menezes, Software implementation of elliptic curve cryptography over binary fields, in *International Workshop on Cryptographic Hardware and Embedded Systems*, pp. 1–24, 2000.

[17] C.-M. Chen, S.-M. Chen, X. Zheng, L. Yan, H. Wang, and H.-M. Sun, Pitfalls in an ECC-based lightweight authentication protocol for low-cost RFID, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 5, no. 4, pp. 642–648, 2014.

[18] N. Jia, S. Liu, Q. Ding, S. Wu, and X. Pan, A new method of encryption algorithm based on chaos and ecc, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 7, no. 3, pp. 637–643, 2016.

[19] C. Yang, J.-S. Pan, C.-Y. Lee, and L. Yan, Reduction of space complexity based on symmetric TMVP, *Electronics Letters*, vol. 51, no. 9, pp. 697–699, 2015.

[20] W. N. Chelton and M. Benaissa, Fast elliptic curve cryptography on FPGA, *IEEE Transactions on Very Large Scale Integration (VLSI) systems*, vol. 16, no. 2, pp. 198–205, 2008.

[21] J.-S. Pan, C.-Y. Lee, and Y. Li, Subquadratic space complexity gaussian normal basis multipliers over g $GF(2^m)$ based on dickson–karatsuba decomposition, *IET Circuits, Devices & Systems*, vol. 9, no. 5, pp. 336–342, 2015.

[22] D. J. Bernstein, T. Lange, and R. R. Farashahi, Binary edwards curves, in *International Workshop on Cryptographic Hardware and Embedded Systems*, pp. 244–265, 2008.

[23] A. P. Fournaris, N. Sklavos, and C. Koulamas, A high speed scalar multiplier for binary edwards curves, in *Proceedings of the Third Workshop on Cryptography and Security in Computing Systems*, pp. 41–44, 2016.
[24] C. F. Kerry and P. D. Gallagher, Digital signature standard (DSS), *FIPS PUB*, pp. 186–4, 2013.
[25] J.-S. Pan, C.-Y. Lee, A. Sghaier, M. Zeghid, and J. Xie, Novel systolization of subquadratic space complexity multipliers based on toeplitz matrix–vector product approach, *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 7, pp. 1614–1622, 2019.
[26] J.-S. Pan, C.-S. Yang, and C.-Y. Lee, Decomposition of symmetric matrix–vector product over $GF(2^m)$, *Electronics Letters*, vol. 53, no. 24, pp. 1568–1570, 2017.
[27] C. W. Chiou, C.-Y. Lee, J.-M. Lin, Y.-C. Yeh, and J.-S. Pan, Low-latency digit-serial dual basis multiplier for lightweight cryptosystems, *IET Information Security*, vol. 11, no. 6, pp. 301–311, 2017.
[28] A. Zakerolhosseini and M. Nikooghadam, Low-power and high-speed design of a versatile bit-serial multiplier in finite fields $GF(2^m)$, *Integration*, vol. 46, no. 2, pp. 211–217, 2013.
[29] B. Sunar and C. K. Koc, Mastrovito multiplier for all trinomials, *IEEE Transactions on Computers*, vol. 48, no. 5, pp. 522–527, 1999.
[30] S. Talapatra, H. Rahaman, and J. Mathew, Low complexity digit serial systolic montgomery multipliers for special class of $GF(2^m)$, *IEEE Transactions on Very Large Scale Integration (VLSI) systems*, vol. 18, no. 5, pp. 847–852, 2009.
[31] J.-S. Pan, P. Song, and C.-S. Yang, Efficient digit-serial modular multiplication algorithm on FPGA, *IET Circuits, Devices & Systems*, vol. 12, no. 5, pp. 662–668, 2018.
[32] J.-S. Pan, C.-Y. Lee, and P. K. Meher, Low-latency digit-serial and digit-parallel systolic multipliers for large binary extension fields, *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, no. 12, pp. 3195–3204, 2013.
[33] M. Morales-Sandoval, C. Feregrino-Uribe, P. Kitsos, and R. Cumplido, Area/performance trade-off analysis of an fpga digit-serial $GF(2^m)$ montgomery multiplier based on lfsr, *Computers & Electrical Engineering*, vol. 39, no. 2, pp. 542–549, 2013.
[34] C.-Y. Lee, C.-S. Yang, B. K. Meher, P. K. Meher, and J.-S. Pan, 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*, vol. 61, no. 11, pp. 3115–3124, 2014.
[35] C.-Y. Lee and P. K. Meher, Subquadratic space-complexity digit-serial multipliers over $GF(2^m)$ using generalized (a, b)-way karatsuba algorithm, *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 62, no. 4, pp. 1091–1098, 2015.