Internet of Things and Cloud Computing
2019; 7(1): 31-38
http://www.sciencepublishinggroup.com/j/iotcc
doi: 10.11648/j.iotcc.20190701.15
ISSN: 2376-7715 (Print); ISSN: 2376-7731 (Online)

Efficient Hardware Implementation of Modular Arithmetic and Group Operation over Prime Field

Sakib Absar[1], Md Selim Hossain[1], Yinan Kong[2]

[1]Department of Electrical and Electronic Engineering (EEE), Khulna University of Engineering & Technology (KUET), Khulna, Bangladesh
[2]School of Engineering, Macquarie University, Sydney, NSW, Australia

Email address:
apuabsar@gmail.com (S. Absar), selim@eee.kuet.ac.bd (M. S. Hossain), yinan.kong@mq.edu.au (Yinan Kong)

To cite this article:
Sakib Absar, Md Selim Hossain, Yinan Kong. Efficient Hardware Implementation of Modular Arithmetic and Group Operation over Prime Field. *Internet of Things and Cloud Computing*. Vol. 7, No. 1, 2019, pp. 31-38. doi: 10.11648/j.iotcc.20190701.15

Received: May 5, 2019; **Accepted**: May 22, 2019; **Published**: June 15, 2019

Abstract: The need for secure communication over the network has increased drastically over recent years, and Elliptic Curve Cryptography (ECC) carries out a significant role in moving secured information. In this work, a hardware implementation of modular arithmetic and group operations over the prime field for an Elliptic Curve Cryptography Processor (ECP) for an efficient security system is proposed. The modular addition or subtraction operation takes only one clock cycle and the modular multiplication, which is designed using the interleaved modular multiplication method, requires 257 clock cycles. For elliptic curve group operation separate point doubling (PD) and point addition (PA) architectures are implemented in Jacobean coordinates. These new architectures are simulated in a Xilinx ISE 14.7. After that, the architectures are implemented in Xilinx Virtex-7 field-programmable gate array (FPGA) with the VHDL language. Proposed modular arithmetic and group operations can be utilized to design an Elliptic Curve Point Multiplication (ECPM).

Keywords: Elliptic Curve Cryptography (ECC), Modular Arithmetic, Elliptic Curve Group Operation, Point Doubling (PD), Point Addition (PA), Field-Programmable Gate Array (FPGA)

1. Introduction

Safety of data has emerged as a crucial factor for preventing unapproved access to websites, personal files, personal payment information, bank account details and personal databases. Cryptography, which allows only the sender and the intended recipient to explore the contents of a message, could provide this required data security. ECC is one of the PKC systems which was proposed for the first time in the mid-1980s by N. Koblitz [1] and V. S. Miller [2]. The security of ECC typically depends on the difficulty of the discrete logarithm problem. IEEE [3] and ANSI [4] have published ECC parameters. ECC usage has appeared to be more productive than other public-key cryptography systems. For instance, a 256-bit ECC can give a similar dimension of safety as a 3072-bit Rivest– Shamir– Adleman RSA [5]. It is observed that less memory and hardware resources are needed for executing elliptic curve cryptography processors. This makes ECC extremely popular for gadgets such as smart cards and cellular phones.

Numerous hardware architectures have been proposed regarding the modular arithmetic operation and elliptic curve group operations over a prime field. There are a lot of architectures built for the modular multiplication operation, as the overall results depend vastly on this operation. Lee et al. implemented the Radix-4 modular multiplication method [6]. The Montgomery modular multiplication method has been applied by the authors [7]. The authors [8] have implemented modular multiplication operation using Booth Radix-4 multiplication and Moore multiplication methods. In the research [9], the authors have proposed a modified interleaved modular multiplication method which has proved to have better performance than most other available designs. For the group operations, both combined and separate architectures for point doubling and point addition over the prime field for 256-bit have been proposed in the research [9, 10]. Hardware implementation results for group operations were presented by the authors [11] over the 192-bit prime field. Point doubling and point addition operations over binary field have been implemented using Xilinx by the authors [12]. The authors [13] have presented a flexible and fast technique for elliptic curve point arithmetic over the prime field. In the research [14], the authors have presented point doubling and point addition results using Montgomery multiplication method.

The prime focus of this paper is building the hardware architecture for efficient modular arithmetic and group operations for elliptic curve cryptography processor. To perform this, hardware architectures of modular addition, subtraction, multiplication along with separate point doubling and point addition operations have been designed. To achieve the best performance, the modified interleaved multiplication method has been used. Pre-computation and parallelization of operations have been applied to the elliptic curve group operations to reduce the delay.

2. Preliminaries

In this section, comprehensive information about elliptic curve cryptography over the prime field, modular arithmetic operations and the elliptic curve group operations have been described. If p is a prime then the prime field GF(p) can be described as the finite field for which elements x, y ∈GF(p) are integers and the range is between 0 and (p - 1) [7].

2.1. Elliptic Curve Cryptography

Elliptic curve cryptography (ECC) sets up keys with the help of the properties of the elliptic curve equation rather than the common strategy for generating keys. ECC can be designed in either a prime field or a binary field, as the security levels provided by both of them are almost the same [10]. An elliptic curve E over a field K, which stands for E = K, can be expressed by the elongated form of the Weierstrass equation [15]

$$\mathrm{E}: \mathrm{y}^2 + \mathrm{a}_1\mathrm{xy} + \mathrm{a}_3\mathrm{y} = \mathrm{x}^3 + \mathrm{a}_2\mathrm{x}^2 + \mathrm{a}_4\mathrm{x} + \mathrm{a}_6 \tag{1}$$

Where the coefficients $\mathrm{a}_1, \mathrm{a}_2, \mathrm{a}_3, \mathrm{a}_4$ and a_6 belong to the field K. There exist two conditions to develop the compressed model of the Weierstrass equation.

Depending on the condition that field K exhibits feature = 2 or 3 and when a_1= 0, the elliptic curve E becomes

$$\mathrm{y}^2 + \mathrm{xy} = \mathrm{x}^3 + \mathrm{ax}^2 + \mathrm{b} \tag{2}$$

This equation is considered as the elliptic curve E over the binary field GF(2^m).

Depending on the condition that the field K exhibits feature ≠ 2 and ≠ 3, the elliptic curve E becomes

$$\mathrm{y}^2 = \mathrm{x}^3 + \mathrm{ax} + \mathrm{b} \tag{3}$$

Where x, y, a, b∈ GF(p) with

$$4\mathrm{a}^3 + 27\mathrm{b}^2 \neq 0 \tag{4}$$

This equation is called the elliptic curve E over the prime field GF(p).

The Weierstrass equation of the elliptic curve is

$$\mathrm{Y}^2 = \mathrm{X}^3 + \mathrm{aXZ}^4 + \mathrm{bZ}^6 \tag{5}$$

2.2. Modular Arithmetic over Prime Field

A finite field, also known as a Galois field, refers to a field in which there exist finitely many components. A finite cyclic group is necessary for a cryptosystem in which the group operations can be efficiently calculated. There are three types of usable finite field: prime field, extension field, and binary field [15, 16].

The modular addition operation usually adds two inputs x, y and subtracts the modulus p from the sum of x, y until the sum Z becomes less than the modulus p.

$$\mathrm{Z} = \mathrm{x} + \mathrm{y}\ (\mathrm{mod}\ p) \tag{6}$$

For modular subtraction, y is inverted bitwise and then added to x with a carry-in set to 1. If the subtraction result becomes negative then the modulus p has to be added to bring the output within the correct range [9].

$$\mathrm{Z} = \mathrm{x} - \mathrm{y}\ (\mathrm{mod}\ p) \tag{7}$$

Modular multiplication is regarded as the most valuable and significant operation over the prime field. The basic modular multiplier operation of two elements is presented as

$$\mathrm{Z} = \mathrm{x} \times \mathrm{y}\ (\mathrm{mod}\ p) \tag{8}$$

2.3. Elliptic Curve Group Operations

The following level in the hierarchy of an elliptic curve cryptosystem is an elliptic curve group operation i.e. point doubling (PD) and point addition (PA). The elements of these are the modular arithmetic operations such as modular multiplication, inversion, squaring, addition, subtraction [17]. In the PA, two distinct points on the elliptic curve are added. For two dissimilar points P = (x_1, y_1) and Q = (x_2, y_2), PA is the point R (R = P + Q) which can be defined as the line on the elliptic curve E that crosses the points P and Q; also the point R represents the reflection of the point about the x-axis. Likewise, the PD is the result of adding a point P to itself which can be defined as R = 2P. If the tangent to the elliptic curve at P is drawn then the projection over the x-axis of the point is defined as R. Both PD and PA can be computed in either a binary field or a prime field [9]. The Jacobian projective model for the Weierstrass equation is expressed by

$$\mathrm{Y}^2 = \mathrm{X}^3 + \mathrm{aXZ}^4 + \mathrm{bZ}^6 \tag{9}$$

Where the point of projective coordinates is represented by P = (X, Y, Z).

For the PD operation, for a point P = (X_1,Y_1,Z_1), the PD in Jacobian projective coordinates for the Koblitz curve R = (X_3,Y_3,Z_3) = 2P can be calculated by [18]

$$X_3 = (3X_1^2)^2 - 8X_1Y_1^2 \tag{10}$$

$$Y_3 = (3X_1^2)(4X_1Y_1^2 - X_3) - 8Y_1^4 \tag{11}$$

$$Z_3 = 2Y_1Z_1 \tag{12}$$

For the PA operation, for two points P = (X_1,Y_1,Z_1) and Q=(X_2,Y_2,Z_2), the PA in Jacobian coordinates for the Koblitz curve R = (X_3, Y_3, Z_3) = P + Q can be calculated by

$$X_3 = A^2 - B^3 - 2X_1Z_2^2B^2 \tag{13}$$

$$Y_3 = A(X_1Z_2^2B^2 - X_3) - Y_1Z_2^3B^3 \tag{14}$$

$$Z_3 = Z_1Z_2B \tag{15}$$

Where

$$A = Y_2Z_1^3 - Y_1Z_2^3 \tag{16}$$

$$B = X_2Z_1^2 - X_1Z_2^2 \tag{17}$$

3. Hardware Architecture Over GF(p)

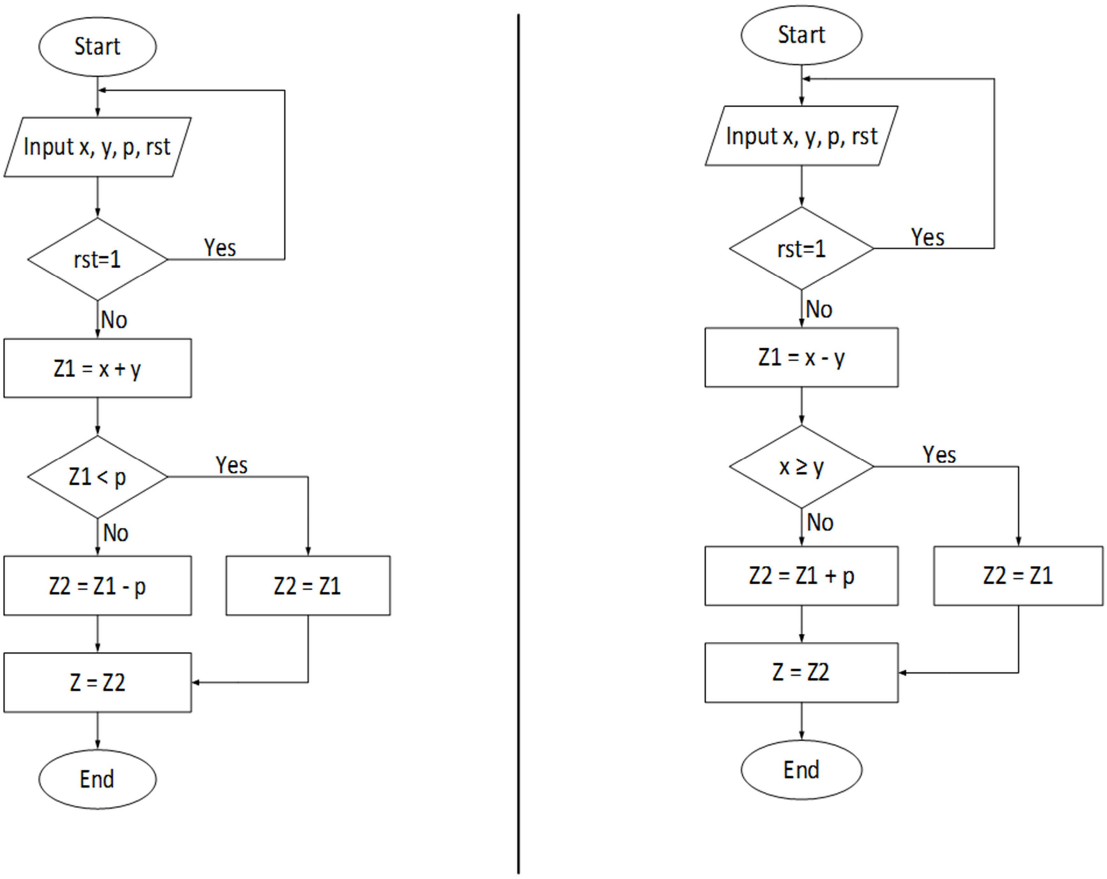
In this section, the proposed architectures for the modular arithmetic operations and elliptic curve group operations are demonstrated. Three architectures have been developed for modular addition, modular subtraction and modular multiplication; another two for group operations, i.e. point doubling and point addition. For all the hardware architectures, the National Institute of Standards and Technology (NIST) standard for $\mathbb{F}_{256}$ has been used. The prime number based on the NIST standards, for 256-bit operations for Koblitz curves over a prime field is

$$p = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$$

$$p = 115792089237316195423570985008687907853269984665640564039457584007908834671663$$

3.1. Hardware Architecture of Modular Arithmetic

The flow chart for the modular addition operation is presented in Figure 1(a). According to the algorithm, two inputs x and y are added. The intermediate result $(x + y)$ could become greater than the predetermined value of modulus$p$. An adder is used that sums up the intermediate result $(x + y)$ to the bitwise-inverted modulus p with the carry-in set to 1 for subtracting the modulus p from the intermediate result. This action can be regarded as two's-complement subtraction. The intermediate output is checked by the carry out of the second adder as to whether it remains in the predetermined range or not. If the sum $(x + y)$ remains in the proper range, the output of the first adder can be considered to be accurate; else, the output of the second adder is right. The condition is checked by using a multiplexer to select whether $(x + y)$ is greater than or equal to the predetermined value of$p$. Only one clock cycle (CC) is required to complete the whole operation. A corresponding hardware architecture for modular addition is illustrated in Figure 2(a).



(a) (b)

Figure 1. *Flow charts for (a) modular addition, (b) modular subtraction.*

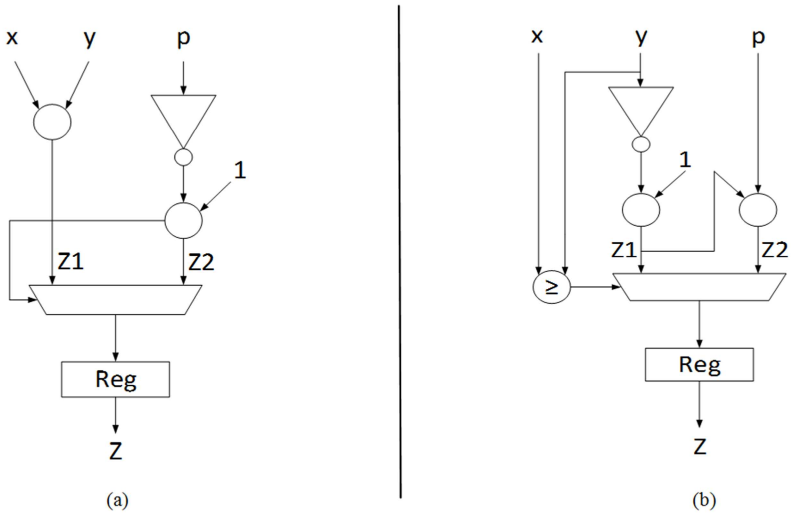


Figure 2. *Hardware architectures for (a) modular addition, (b) modular subtraction*

The flow chart for the modular subtraction operation is illustrated in Figure 1(b). The modular subtraction operation is achieved in a similar process to modular addition. With a carry-in set to 1, y is bitwise inverted and added to x to perform the modular subtraction operation. Modulus p will be added if the result becomes negative, thus producing an output in the exact range between 0 and $p - 1$. This operation also requires only one cycle to be completed. A corresponding hardware architecture for modular subtraction is presented in Figure 2(b).

The algorithm for modular multiplication is illustrated in Algorithm 1 which is based on the modified interleaved multiplication method. The critical path consists of only gates, an adder and a subtractor, which makes this modular multiplication algorithm a thoroughly efficient one. If m represents the bit length of the operands A, B or p, this approach needs (m +1) cycles to get the outcome. So, for a 256-bit operation, a total of 257 CCs are needed. The modular squaring operation also needs 257 CCs, differs from the modular multiplication operation only in terms of inputs, as the inputs of the squarer need to be identical but, for the multiplier, the inputs can be different.

```
Algorithm 1: Modular multiplication algorithm in GF($p$)

Input: Prime $p$ and $A, B \in [1, p-1]$
Output: $C = (A*B) \bmod p$
1.    $C = 0; p_2 = 2 * p$ ( pre-computed);
2.1   for $i = m-1$ down to 0 do
2.2   $c_1 = C; c_2 = 2 * c_1$ ( left-shift operation);
2.3   $i_1 = A[i] * B$ ( and-gate operation);
2.4   $c_3 = c_2 + i_{1s}; c_4 = c_3 - p; c_5 = c_3 - p_2\ (p_2 = 2p)$;
2.5   if $c_3 \geq p$ then $c_6 = c_4$;
2.6   else if $c_3 \geq p_2$ then $c_6 = c_5$; else $c_6 = c_3$;
      end if $C = c_6$
2.7   end for
3.    Return $C$;
```

3.2. Hardware Architecture of Group Operation

The elliptic curve group operations PD and PA are designed using a finite-field modular arithmetic (FFMA) unit. Separate PD and PA architectures are designed for the elliptic curve group operations.

The proposed hardware architecture for point doubling is presented in Figure 3. The architecture is built based on the equations of PD over GF(p) in Jacobian projective coordinates for the Koblitz curve. Pre-computation, balancing of architecture and parallelization of operations have been applied to reduce the total number of clock cycles (CCs), power consumption, longest path and to increase the speed of operation for more adequate performance. For example, at level 1, two multiplications and one squaring are operated in parallel. So only 257 CCs are required instead of 3×257 CCs, i.e. the number of CCs needed in level 1 is reduced to one-third of the number of CCs if parallelization was not applied. Similarly, in level 3, six additions and one multiplication are operated in parallel. As a result, instead of 257 + (6 × 1) CCs, only 257 CCs are required in level 3. Modular inversion has been avoided as it is the most costly operation. The overall latency required for PD is $4m$+3, where m is the latency of multiplication. The overall architecture requires 3 multipliers, 4 squarers, 9 adders and 3 subtractors. Therefore the combined cost of PD is 3MUL+4SQ+9ADD+3SUB. Only seven levels are required to complete the total PDBL operation. The required total number of clock cycles is (257+257+257+1+1+257+1) = 1031 CCs.

Based on the equations of PA over GF(p) for the Koblitz curve in Jacobian projective coordinates, a hardware architecture for point addition is proposed and is shown in Figure 4. Pre-computation, balancing of architecture and parallelization of operations have also been used for the point addition operation for achieving better performance. For example, five multiplications are operated in parallel in level 2. So only 257 CCs are required instead of 5×257 CCs i.e. the number of CCs needed in level 2 is reduced to one-fifth the number of CCs if parallelization was not applied.

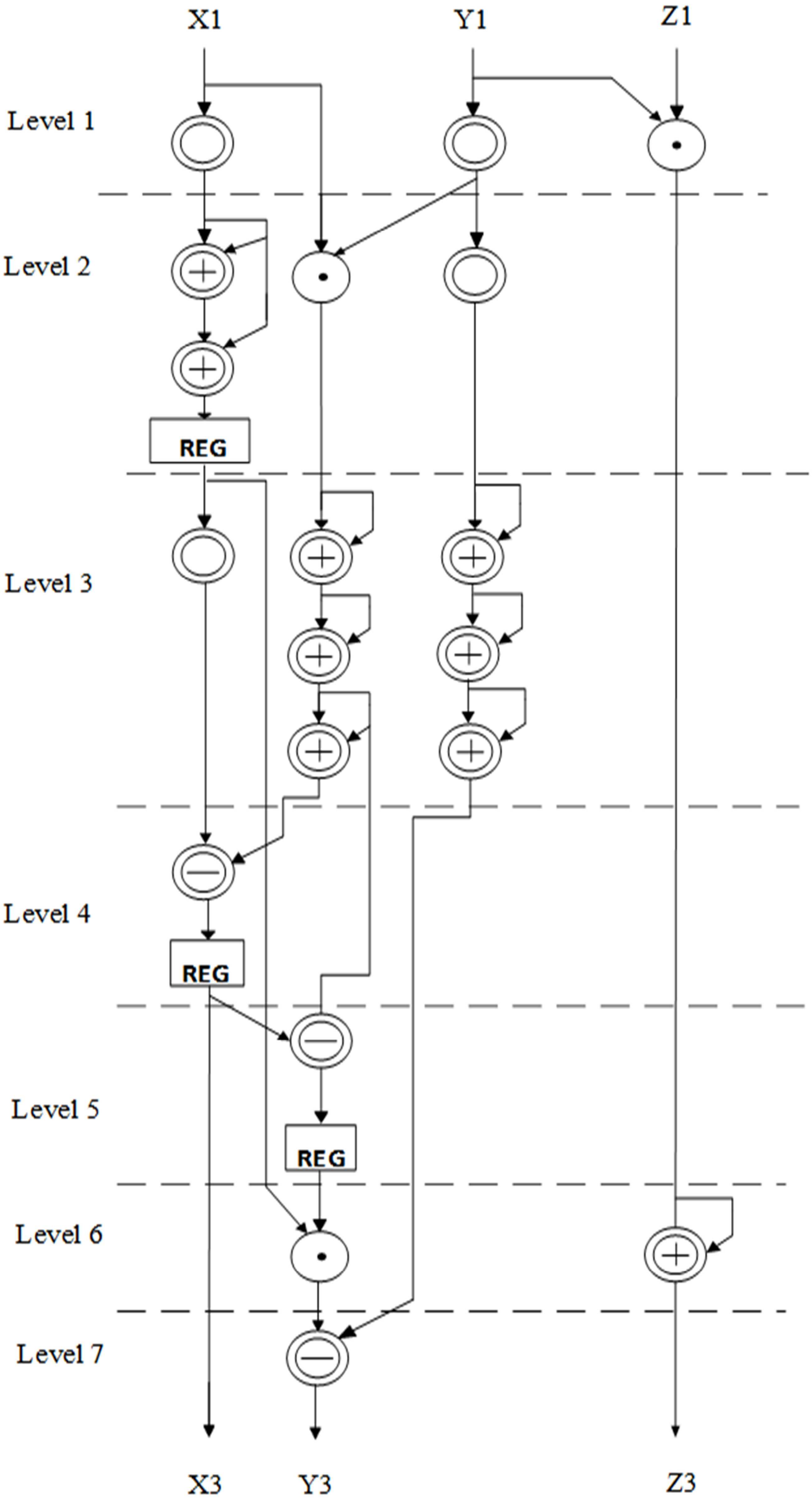


Figure 3. *Proposed architecture for the point doubling (PD) operation.*

The overall latency required for PA is 6*m*+4, where *m* is the latency of multiplication. The overall architecture requires 12 multipliers, 4 squarers, 1 adder and 6 subtractors. Hence the combined cost of PA is 12MUL+4SQ+1ADD+6SUB. Only 10 levels are required to complete the total PD operation. The required total number of clock cycles is (257+257+257+257+257+1+1+1+257+1) = 1546 CCs.

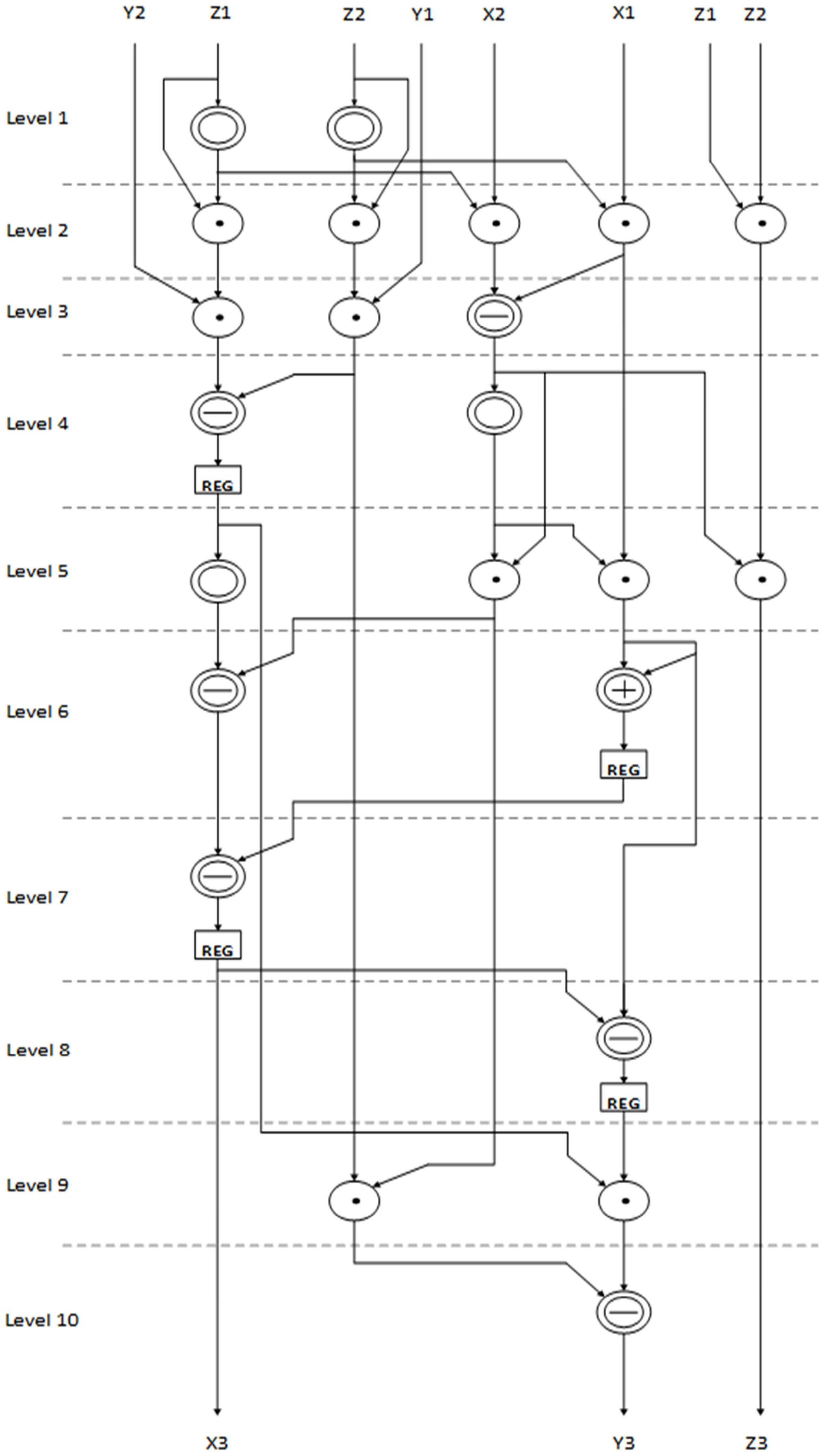


Figure 4. *Proposed architecture for the point addition (PA) operation.*

4. Results and Analysis

This passage shows the results of implementations for the proposed hardware architectures. The proposed architectures have been simulated using ModelSim PE and synthesized using a Xilinx ISE 14.7 for an FPGA board of the Virtex-7 family with device code xc7vx485t, with an ultimate objective of high speed. The simulation results have been verified with the help of Maple software.

NIST prime p -256 was used for all the hardware implementations of this work, and the Jacobian coordinate system has been used in them. Simulation results for the modular arithmetic operations are given in Table 1. The proposed modular adder and subtractor require only 1 clock cycle each. The modular multiplier is designed using the interleaved modular multiplication method which requires 257 clock cycles. This modular multiplier need 1470 slice LUTs and 514 LUT-FF pairs with a total time of 2.09 μs.

Table 1. *Simulation results for modular arithmetic operations for ECP.*

Module	Modular Addition	Modular Subtraction	Modular Multiplication
Required no. of Clock cycles	1	1	257
Minimum Time Period (ns)	5.66	5.67	8.14
Maximum Frequency (MHz)	176.83	176.46	122.86
Total Time Required (ns)	5.66	5.67	2091.72
Required Slice LUTs	768	1023	1470
Required LUT FF-pairs	0	0	514
Required bonded IOBs	771	771	772

The results of the hardware implementation of the group operations are illustrated in Table 2. The point doubling and point addition operations were implemented using pre-computation and parallelization of operations for reducing the delay. For the PD operation, the number of required LUT-FF pairs is 6644, the number of required slice LUTs is 19095, the number of required slice registers is 6701 and the delay is 8.48 μs. For the PA operation, the number of required LUT-FF pairs is 9602, the number of required slice LUTs is 31393, the number of required slice registers is 10048 and the delay is 12.72 μs.

Table 3 represents a relative comparison of proposed work with some similar works over GF(p). The hardware architecture for PD operation described in this paper needs only 8.48 μs at a frequency of 121.62 MHz with 1031 KCycles whereas the hardware architecture for PA operation needs only 12.72 μs at a frequency of 121.54 MHz with 1546 KCycles. Comparing the results with other available works present in this literature; it can be seen that the hardware architectures described in this paper for the group operations provide better timing results than most other available works.

Table 2. *Simulation results for elliptic curve group operations for ECP.*

Module	PDBL	PADD
Required no. of clock cycles	1031	1546
Minimum Time Period (ns)	8.22	8.23
Maximum Frequency (MHz)	121.62	121.54
Total Time Required (ns)	8476.88	12718.94
Required Slice Registers	6701	10048
Required Slice LUTs	19095	31393
RequiredLUT FF-pairs	6644	9602
Required bonded IOBs	1540	2308

Table 3. *Comparison between proposed group operation designs and similar work over GF(p).*

Work	Platform	Bit length	Group operation	Time (μs) @f (MHz)	KCycles
This work	Virtex-7	$\mathbb{F}_{256}$	PD	8.48@121.62	1031
			PA	12.72@121.54	1546
[19]	Virtex-4	$\mathbb{F}_{256}$	PD	17.20@60	1020
			PA	18.40@60	1089
[20]	Virtex-7	$\mathbb{F}_{256}$	PD	8.49@121.62	1032
			PA	12.72@121.54	1547
[7]	Kintex-7	$\mathbb{F}_{256}$	PD	7.65@146.40	1285
			PA	7.57@146.40	1283
[21]	Virtex-2 pro	$\mathbb{F}_{256}$	PD	8.31@39.46	328
			PA	13.33@39.46	526

5. Conclusion

A high-performance hardware architecture of modular arithmetic and group operations for elliptic curve cryptosystem over the prime field $\mathbb{F}_{256}$ is implemented considering the trade-off between area and time. The Jacobian coordinate system has been used for implementing the architecture to avoid the expensive modular inversion procedure. Separate modular adder and subtractor architectures were implemented. Both of them need only one clock cycle. For the modular multiplication operation, modified interleaved modular multiplication is used because the intermediate result in this technique is only one or two bit larger than the operands, as the intermediate result is always reduced by taking the modulus. This multiplication needs 257 clock cycles and a delay of 2.09 µs to operate which is faster than most other related architectures. In order to achieve the best performance, pre-computation, balancing of architecture and parallelization of operations have been applied for implementing the architectures of point doubling and point addition. As a result, the number of total levels and the required logic stages get reduced. The proposed design of PD takes only 8.48 µs with 1031 clock cycles whereas PA takes only 12.72 µs with 1546 clock cycles. Analyzing the performance and comparing with other similar works in table 3, it is seen that these results are better than most other available architectures in the literature. All the proposed architectures are synthesized using a Xilinx Virtex-7 FPGA. These architectures are used in elliptic curve scalar multiplication or elliptic curve point multiplication to build an overall efficient hardware architecture for the elliptic curve cryptosystem over the prime field $\mathbb{F}_{256}$.

Acknowledgements

The authors would like to express their thanks to the Committee for Advanced Studies and Research (CASR), Khulna University of Engineering & Technology (KUET), Khulna-9203, Bangladesh.

References

[1] Koblitz, N.: ‘Elliptic curve cryptosystems’, Math. Comput., 1987, vol. 48, pp. 203–209.

[2] Miller, V. S.: ‘Use of elliptic curves in cryptography’. In Proc. CRYPTO 1985, 1986, pp. 417–426.

[3] “IEEE standard specifications for public-key cryptography,” IEEE Std 1363-2000, pp. 1–228, Aug. 2000.

[4] X 9.62 public key cryptography for the financial services industry: Elliptic curve digital signature algorithm (ECDSA), “American National Standards Institute,” 1999.

[5] Rivest, R. L., Shamir, A., Adleman, L.: ‘A method for obtaining digital signatures and public-key cryptosystems’, *Commun. ACM*, 1978, 21, (2), pp. 120–126.

[6] J. W. Lee, S.-C. Chung, H. C. Chang, and C. Y. Lee: “Efficient power analysis-resistant dual-field elliptic curve cryptographic processor using heterogeneous dual-processing-element architecture,” *IEEE Trans. VLSI Syst.*, vol. 22, pp. 49-61, Jan. 2014.

[7] M. S. Hossain, Y. Kong: ‘FPGA-Based Efficient Modular Multiplication for Elliptic Curve Cryptography’, in ITNAC, pp. 191-195, Nov. 2015.

[8] M. R. Hossain, M. S. Hossain: ‘Efficient FPGA implementation of modular arithmetic for elliptic curve cryptography’, *ECCE 2019*, Cox’s Bazar, Bangladesh, 2019, pp. 1-6.

[9] M. S. Hossain, “High-Performance Hardware Implementation of Elliptic Curve Cryptography”, Research Book, Doctor of Philosophy, Department of Engineering, Faculty of Science and Engineering, Macquarie University, Sydney, Australia, Jun. 2017.

[10] M. S. Hossain, Y. Kong, E. Saeedi, N. C. Vayalil: “High-Performance elliptic curve cryptography processor over NIST prime fields,” IET Computers & Digital Techniques, 2017, vol. 11, no. 1, pp. 33-42.

[11] M. Jaiswal, K. Lata: ‘Hardware implementation of text encryption using elliptic curve cryptography over 192 bit prime field’, *ICACCI 2018,* pp. 343-349, Sept. 2018.

[12] Megha M. Panchbhai, U. S. Ghodeswar: ‘Implementation of point addition & point doubling for elliptic curve’ IEEE ICCSP 2015, Melmaruvathur, India, 2015, pp. 746-749.

[13] P. Longa, A. Miri: ‘Fast and flexible elliptic curve point arithmetic over prime fields’, *IEEE Trans.,* vol. 57, pp. 289-302, Mar. 2008.

[14] A. Satoh, K. Takano: ‘A scalable dual-field elliptic curve cryptography processor’, *IEEE Trans.,* vol. 52, pp. 449-460, Apr. 2003.

[15] D. Hankerson, A. J. Menezes, and S. Vanstone, ‘Guide to Elliptic Curve Cryptography’. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2003.

[16] G. Sutter, J. Deschamps, and J. Imana, “Efficient elliptic curve point multiplication using digit-serial binary field operations,” IEEE Trans. Ind. Electron., vol. 60, no. 1, pp. 217–225, Jan. 2013.

[17] M. S. Hossain, E. Saeedi, Y. Kong: ‘Parallel point-multiplication architecture using combined group operations for high-speed cryptographic applications’, PLOS ONE, May 2017.

[18] G. Orlando and C. Paar, “A high-performance reconfigurable elliptic curve processor for GF(2m),” in Proc. CHES, 2000, pp. 41–56.

[19] K. Ananyi, H. Alrimeih, D. Rakhmatov: ‘Flexible hardware processor for elliptic curve cryptography over NIST prime fields’, *IEEE Trans. VLSI Syst.*, vol. 17, no. 8, pp. 1099-1112, Aug. 2009.

[20] M. S. Rahman, M. S. Hossain, E. H. Rahat, D. R. Dipta, H. M. R. Faruque, F. K. Fattah: ‘Efficient hardware implementation of 256-bit ECC processor over prime field’, *ECCE 2019*, Cox'sBazar, Bangladesh, 2019, pp. 1-6

[21] C. J. McIvor, M. Mchoone, J. V. McCanny, “Hardware elliptic curve cryptography processor over GF (*p*),”*IEEE Trans. Circuits Syst. I*, 2006, vol. 53, pp. 1946-1957.