International Journal of Trend in Research and Development, Volume 2(6), ISSN: 2394-9333
www.ijtrd.com

Lattice based Multiplier for WSN Applications for ECC

[1]S.Pavithra and [2]S.Baskar,
[1]PG Scholar, VLSI Design and [2]Assistant professor,
[1,2]Department of Electronics and Communication Engineering,
Angel College of Engineering and Technology, Tirupur, Tamil Nadu, India.

Abstract- Elliptic curve based cryptosystem (ECC) is an efficient public key cryptosystem, which is more suitable for limited environments with the keys of smaller size. The performance of elliptic curve cryptosystem heavily depends on point multiplication. Elliptic curve cryptography is especially useful for wireless sensor network (WSN), which enables wireless devices to perform secure communication efficiently and establishes secure end to end connections. This paper gives an introduction to ECC and comparative presents the study of methods for lattice multiplication operation. Lattice multiplication is performed in the binary method as this improves the speed and accuracy of the multiplication. The proposed design involves significantly less delay and complexities when compared to traditional multipliers. The presented simulation results show the validation of our method and analysis.

Keywords: *Elliptic Curve Cryptography (ECC), Wireless Sensor Network (WSN), Lattice Multiplication.*

I. INTRODUCTION

The fast progress in wireless communication systems, personal communication systems, and smartcard technologies has brought new opportunities and challenges to be met by engineers and researchers functioning on the protection aspects of the new communication technologies. Wireless sensor network (WSN) consists of a large number of sensor nodes that are able to collect and broadcast data in areas where ordinary networks are unsuitable for environmental and/or strategic reasons. There are with limited resources and one or more base station. The base station is a more powerful node that connect the sensor nodes to the world. WSN are widely used for monitoring purpose and provide the information about the monitored area or device. Now-a-days the WSN‘s are also used to control the monitoring device. The elliptic curve cryptosystem has a wide range of application in the wireless communication, it have a performance like high security, high speed and low bandwidth.

In 1985, Elliptic curve cryptography was popular in the data encryption and decryption. Elliptic curve cryptography offers safe and capable solutions for the new communication technologies. The ECC deals with harder problems efficiently when compared with the other cryptosystem. It is gaining wide acceptance as an alternative to the usual public key cryptosystem such as RSA, DSA. The advantage of using the finite group of elliptic curve (EC) is that its discrete logarithm problem is believed to be harder than the discrete logarithm problem for the multiplication group of a finite field. Another advantage that makes EC more striking is the possibility of optimizing the arithmetic operations in the underlying field. The ECC is used in cellular communication, as it provides a smaller key size with same level of security of other cryptosystems. It is used in limited environments like, PDA, sensor networking, etc. The mathematical operations of ECC is characterized over the elliptic curve $y2 = x3 + ax + b$, where $4a3+27b2 \neq 0$. Each value of the ‘a’ and ‘b’ gives various elliptic curves. The public key is a point in the curve and the private key is a random number. The public key is attained by multiplying the private key with the generator point G in the curve. The generator point G, the curve parameters ‘a’ and ‘b’, together with few more constants represent the domain parameter of ECC. In ECC, main operations such as key agreement, signature generation, signing and verification involve scalar multiplication. The speed of scalar multiplication plays a significant role in the efficiency of whole system. So the Fast multiplication is particularly more fundamental for some environments such as central servers, where large numbers of key agreements or signature generations occur, and in handheld devices with low computational power. Because of such importance of scalar multiplication, numerous methods have been developed, such as binary method, signed binary method, sliding window method.

In this paper, we trend to propose a new multiplication method called Lattice multiplication that relies on the simple binary method. This new method not solely reduces the total number of point addition, but also reduces the number of point doubling. The method is straightforward and

particularly appropriate for low time delay. The proposed methods can also be analyzed from a security purpose of view. By considering state of art the attacks on public-key cryptosystems, we tend to prove that the proposed set is safe to use in practice for each elliptic curve cryptography and RSA cryptosystems.

II. ARRAY MULTIPLICATION

Multiplication circuit becomes quick by array multiplier method. A multiplication technique in which an array of identical cells produces new partial product and accumulation of it at constant time.

- Multiplication of decimal A ($0ba_3\ a_2\ a_1\ a_0$)
- Multiplicand (Y) by the decimal B ($0bb_3\ b_2\ b_1\ b0$)
- Multiplier (X)
- An n-bit × n- bit multiplication done in steps.

For example consider A ($0ba_3\ a_2\ a_1\ a_0$) be 0b1011 and B ($0bb_3\ b_2\ b_1\ b_0$) be 0b0101.

- Step A: First, 0b1011 is copied by each bit of 0b0l01 to get the product by LSB (=1).
- Step B: then 0b1011 is taken as 0b0000 to get the product by next to LSB (= 0).
- Step C: then 0b1011 is copied by each bit of 0b0l01 to get the product by next to next of LSB.
- Step D: then 0b1011 is taken as 0b0000 to get the product by MSB (=0).
- The four results are added to get the final result.

Note that each successive line is shifted to the left one position to the left to account for the differing place values of the bits in the second input.

- Step A: n × n Partial product generation using array of AND gates by multiplying $X_0.Y_0$, $X_0.Y_1$, …, up to Xn−1. Yn−1 in parallel at the same time.

n = 4 in the example of multiplying 0b1011 with 0b0l01.

- Step B: Use adders to add the partial products at the n-levels. Note that each level m partial

Product X $X_m.Y_0$, $X_m.Y_1$, …, up to X_1. Yn–1 is shifted to the left one position to the left to account for the differing place values of the bits in the second input

- Step C: Generating final result using two-bit operand adders.

The general format of the array multiplier of various bit is given in the Fig 1, Fig 2, Fig 3.

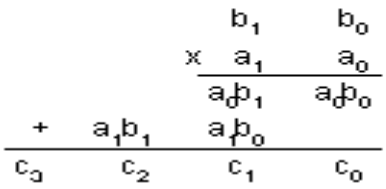

Figure 1: 2-Bit Array Multiplier

Figure 2: 4-Bit Array Multiplier

The example of 4-bit array multiplier which is explained above is given.

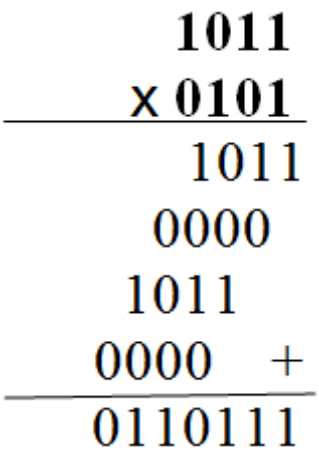

Figure 3: Example of 4-Bit Array Multiplier

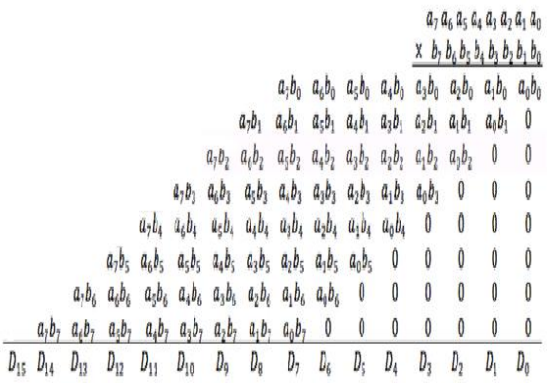

Figure 4: 8-bit multiplier

III. LATTICE MULTIPLICATION

Lattice Multiplication is the method for multiplying bigger numbers or for carrying out complex multiplication. It is algorithmically identical to the traditional long multiplication method, but breaks the process into smaller steps.

In this approach, a lattice is first constructed, sized to fit the numbers being multiplied.

- If we are multiplying an m-digit number by an n-digit number, the size of the lattice is $m \times n$.
- The multiplicand is placed along the top of the lattice so that each digit is the header for one column of cells (the most significant digit is put at the left).
- The multiplier is placed along the right side of the lattice so that each digit is a header for one row of cells (the most significant digit is put at the top).
- Illustrated above is the lattice configuration for computing 948×827.

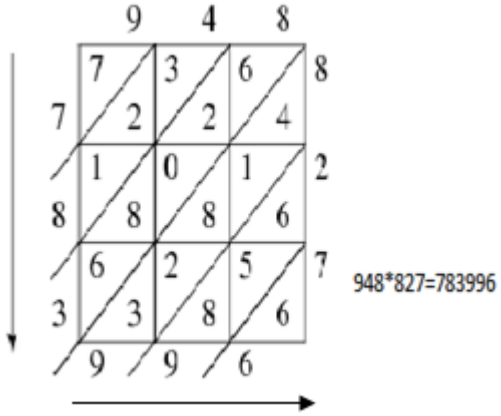

Figure 5: Example of Lattice Multiplication

Before the actual multiplication can begin, lines must be drawn for every diagonal path in the lattice from upper right to lower left to bisect each cell. There will be 5 diagonals for our 3×3 lattice array. Reading the digits down the left side and then towards the right on the bottom to generate the final answer We use the lattice multiplication with the binary number (i.e.) the binary lattice multiplication.

IV. BINARY LATTICE MULTIPLICATION

As in decimal system, the multiplication of binary numbers is carried out by multiplying the multiplicand by one bit of the multiplier at a time and the result of the partial product for each bit is placed in such a manner that the LSB is under the corresponding multiplier bit. Finally the partial products are added to get the complete product. The placement of the numbers is similar to the lattice multiplication and the final product is obtained by reading from the down the left side to the right.

Consider the multiplication of the decimal numbers A (a_1 a_0) and B (b_1 b_0), in their binary representation.

- If we are multiplying an m-bit number by an n-bit number, the size of the lattice is $m \times n$.
- The multiplicand A (a_1 a_0) is placed along the top of the lattice so that each digit is the header for one column of cells (the most significant digit is put at the left).
- The multiplier B (b_1 b_0) is placed along the right side of the lattice so that each digit is a header for one row of cells (the most significant digit is put at the top).
- The product of a_0 x b_0= q_0 is placed in the respective intersection box of the row b_0 and column a_0.
- Similar step is done for a_1 x b_0 = q_1, a_0 x b_1 = q_2, a_1 x b_1 = q_3 and the products are placed in the corresponding boxes.

Note that the multiplication of the number is done by array of AND gates.

- The lines must be drawn for every diagonal path in the lattice from upper right to lower left to bisect each cell.
- The binary number in the diagonal are added using the half adder and full adder to get partial product.

The general format for multiplication of A (a_1 a_0) and B (b_1 b_0) is given

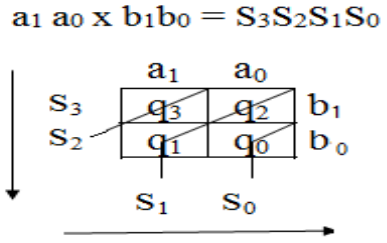

Figure 6: General Format

The final answer $S_3S_2S_1S_0$ is obtained by reading downwards from left to right at the bottom. The method can be used for any bit number for 4-bit, 8-bit and so on.

This is proven by an example; consider the multiplication of the decimal 1 X 3 in their 2-bit binary format 01 X 11. The answer is in 4-bit 0011 whose decimal value is 3.

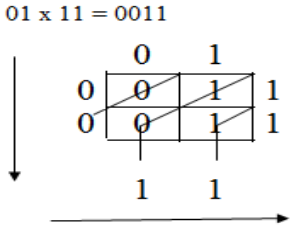

Figure 7: 2-Bit Binary Lattice Multiplier

This method for multiplication is used for multiplying the 4-bit numbers. Here given the multiplication of decimal number 9 X 12 = 108. In their binary format 1001 X 1100 = 01101100 (108) which is of 8-bit.

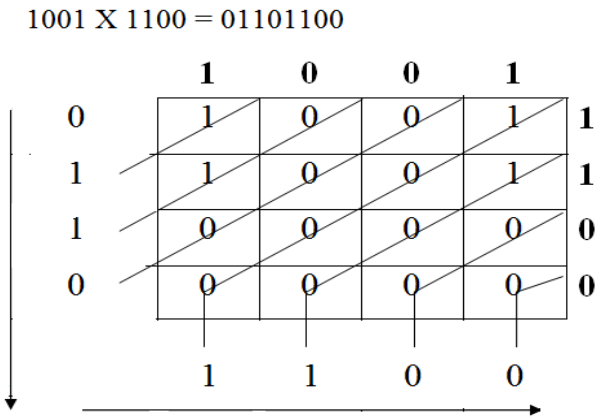

Figure 8: 4-Bit Binary Lattice Multiplier

Similarly for multiplying the 8-bit numbers 00110100 (52) X 00110100 (52) = 000010101001000 (2704). The answer 2704 is in 16-bit.

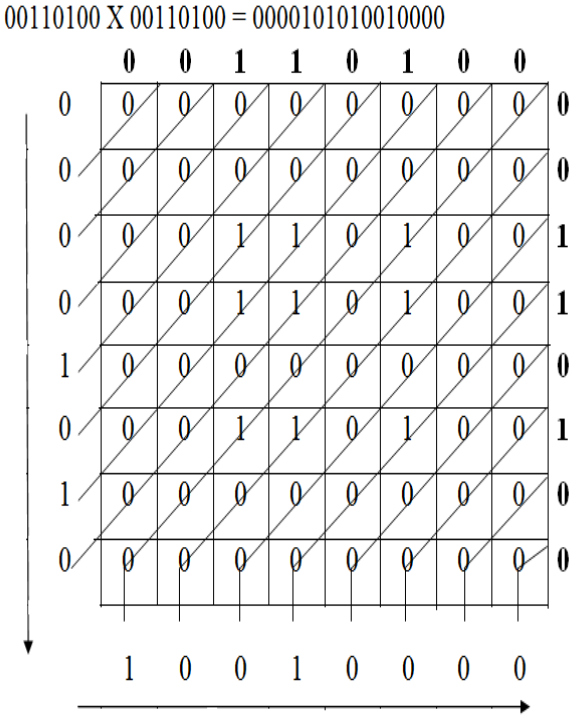

Figure 9: 8-Bit Binary Lattice Multiplier

V. RESULTS ANALYSIS

In this paper, the traditional array multiplier procedure and the proposed binary lattice multiplier process are analyzed. As proven within the (As proven within the figures which is listed in the paper) the number of LUT's, memory utilization and timing are reduced in proposed due to less complexity of adder circuit within the design. The proposed method outperforms the traditional architecture indicates that the proposed process increased the combinational direction delay when in comparison with conventional approach.

Our design has been implemented in Verilog, simulated and synthesized utilizing the Xilinx ISE Design Suite 13.2 device for supply voltage levels from 1.2V to 2.5 V. The proposed binary lattice multiplication for 2-bit,4-bit and 8-bit have been simulated and the result is given.

1) *2-bit multiplier*:

Design Statistics

IOs : 10

Cell Usage:

BELS : 3

GND : 1

LUT2 : 2

IO Buffers : 9

IBUF : 3

OBUF : 6

==========

Number of Slices: 1 out of 960 0%

Number of 4 input LUTs: 2 out of 1920 0%

Number of IOs: 10

Number of bonded IOBs: 9 out of 66 13%

Maximum combinational path delay: 5.776ns

Total REAL time to Xst completion: 2.00 secs

Total CPU time to Xst completion: 2.50 secs

Total memory usage is 190300 kilobytes

Figure 10: Analysis of 2-Bit Binary Lattice Multiplier

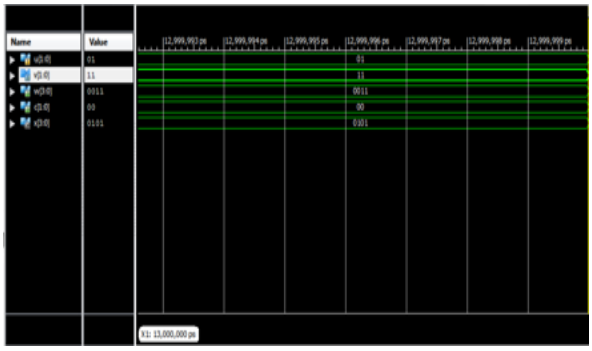

Figure 11: Simulation Result of 2-Bit Binary Lattice Multiplier

2) 4-bit multiplier:

Design Statistics
IOs : 22

Cell Usage:
BELS : 5
GND : 1
LUT2 : 4
IO Buffers : 19
IBUF : 5
OBUF : 14

==========

Number of Slices: 2 Out of 960 0%
Number of 4 input LUTs: 4 Out of 1920 0%
Number of IOs: 22
Number of bonded IOBs: 19 out of 66 28%

Maximum combinational path delay: 5.895ns

Total REAL time to Xst completion: 2.00 secs
Total CPU time to Xst completion: 2.65 secs

Total memory usage is 191324 kilobytes

Fig 12: Analysis of 4-bit binary lattice multiplier

Fig 13: Simulation result of 4-bit binary lattice multiplier

3) 8-bit multiplier:

Design Statistics
IOs : 46

Cell Usage :
BELS : 9
GND : 1
LUT2 : 8
IO Buffers : 39
IBUF : 9
OBUF : 30

==========

Number of Slices: 4 out of 960 0%
Number of 4 input LUTs: 8 out of 1920 0%
Number of IOs: 46
Number of bonded IOBs: 39 out of 66 59%

Maximum combinational path delay: 6.039ns

Total REAL time to Xst completion: 3.00 secs
Total CPU time to Xst completion: 2.81 secs

Total memory usage is 193372 kilobytes

Fig 14: Analysis of 8-bit binary lattice multiplier

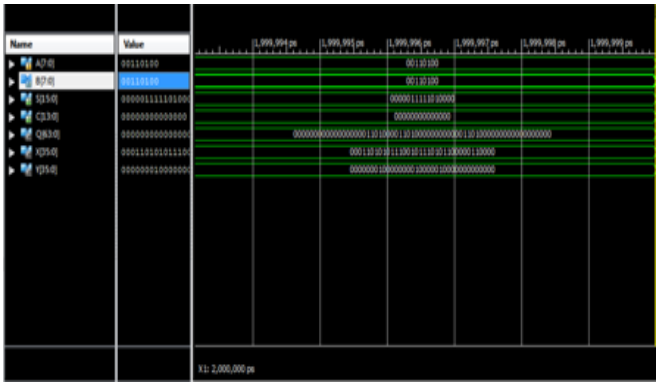

Fig 15: Simulation result of 8-bit binary lattice multiplier

From the above results binary lattice multiplication have less delay time compare to the array multiplier. The highest speed will also be performed with the new architecture in designated configuration.

VI. COMPARISON

The comparison of binary lattice multiplication method with the normal array multiplier is shown in the Table I. The various parameters are compared

which gives binary lattice multiplication is more efficient.

Table 1: Comparison of Multiplication Methods

		ARRAY	BINARY LATTICE
Delay	2-BIT	5.942ns	5.776ns
	4-BIT	12.342ns	5.895ns
	8-BIT	42.082ns	6.039ns
BELS(Basic logic elements)	2-BIT	4	3
	4-BIT	28	5
	8-BIT	131	9
Number of Slices	2-BIT	2 out of 960	1 out of 960
	4-BIT	14 out of 960	2 out of 960
	8-BIT	71 out of 960	4 out of 960

CONCLUSION

Elliptic curve cryptosystem becomes to be the cryptosystem for the future. One way to improve the performance of such cryptosystem is to use a competent method for point multiplication which is the most time consuming operation. Here we proposed a method for point multiplication based on binary method that articulate the idea of lattice multiplication. The proposed method not only reduces the complexity of addition, but also reduces delay time. One of the important advantages of this method is that, we can make several variation of the method by only changing the base. Our method is a very suitable tool for embedded devices such as WSNs. A thorough analysis and simulation based on evaluations will show that the proposed solution does speed up the computation of multiplication on the elliptic curve. The design includes significantly less prolongs than the present structure.

Reference

[1] Alka Sawlikar, Point Multiplication Methods for Elliptic curve Cryptography, International Journal of Engineering and Innovative Technology (IJEIT), January 2012.

[2] Md. Rafiqul Islam, Md. Sajjadul Hasan, Ikhtear Sharif Muhammad Asaduzzaman, A New Point Multiplication Method for Elliptic Curve Cryptography Using Modified Base Representation, International Journal of The Computer, the Internet and Management Vol.16. N.o.2 (May-August, 2008) pp 9-16.

[3] Rahul Nimje, Sharda Mungale, Design of Arithmetic Unit for High Speed Performance Using Vedic Mathematics, IJERA ISSN: 2248-9622 ,ICIAC-12-13th April 2014.

[4] Sandesh S. Saokar, R. M. Banakar, Saroja Siddamal, "High Speed Signed Multiplier for Digital Signal Processing Applications" 2012 IEEE. IEEE International Conference on Signal Processing, Computing and Control (ISPCC), 2012.

[5] G. N. Purohit, Asmita Singh Rawat, Fast Scalar Multiplication in ECC Using The Multi base Number System.

[6] Abhishek Gupta, Arithmetic Unit Implementation Using Delay Optimized Vedic Multiplier with BIST Capability, International Journal of Engineering and Innovative Technology (IJEIT) Volume 1, Issue 5, May 2012.

[7] M. Tian, Y. Wang, S. Xu, "An Efficient Elliptic Curves Scalar Multiplication Algorithm Suitable for Wireles Network" Second International Conference on Networks Security, Wireless Communication and trusted Computing, 2010, pp. 95-98, IEEE Computer Society 2010.

[8] P. K. Mishra, V. S.Dimitrov. E_cient Quintuple Formulas for Elliptic Curves and E_cient Scalar Multiplication Using Multibase Number Rep-resentation. Springer-Verlag, 2007, volume 4779, pages 390-406.

[9] P.Longa (2007): Accelerating the scalar multiplication on Elliptic curve Cryptosystems over prime _elds. Master thesis University Of Ottawa,http://patriclonga.bravehost.com/publications.html.

[10] K.W. Wong, Edward, C.W.Lee, L.M.Cheng, Xiaofeng Liao,(2006), Fast Scalar Multiplication using new Double Base Chain and Point Halving,Applied Mathematics and Computation.

[11] M. Ciet, M. Joye, K. Lauter and P.L. Montgomery,Trading inversions for

multiplications in elliptic curve cryptography, Designs, Codes, and Cryptography, 39, 189-206,(2006).

[12] V. Dimitrov, L. Imbert and P.K. Mishra,(2005),E_cient and Secure Elliptic Curve Point Multiplication using Double-Base Chains,Advances in Cryptology - ASIACRYPT'05, LNCS Vol. 3788, pp. 59-78, Springer-Verlag,2005.

[13] M.Ciet and F.Sica (2005) An Analysis of Double base Number system and a sub linear scalar multiplication Algorithm. LNCS Vol.3715 pp. 171-182.Springer Verlag.

[14] V.S.Dimitrov, L.Imbert, and P.K.Mishra,(2005) Fast elliptic Curve Point Multiplication using Double-Base Chain, Cryptology ePrint Archive , Report 2005/069.

[15] W. Stalling (2003), Cryptography and Network Security, Prentice Hall, New Jersey, USA, Third Edition, Chapter 10.

[16] M. Ciet, M. Joye, K. Lauter, P.L. Montgomery,(2003) Trading Inver-sions for Multiplications in Elliptic Curve Cryptography, Cryptology ePrint Archive, Report 2003/257 . Also to appear in Design, Codes and Cryptography.

[17] K. Okeya and T. Takagi, \The Width-w NAF Method Provides Small Memory and Fast Elliptic Scalar Multiplication Secure against Side Channel Attacks," Topics in Cryptology, The Cryptographers' Track at the RSA Conference 2003 (CT-RSA 2003), LNCS 2612, pp. 328{342,2003}.

[18] E. Al-Daoud, R, mahmod, Md. Rushdan, A. Kilicman (2002),“A new addition formula for Elliptic curve over GF (2n)”,IEEE Transactions on Computers, vol. 51, no. 8, pp. 972-975,Aug.

[19] K. Okeya and T. Takagi, \The Width-w NAF Method Provides Small Memory and Fast Elliptic Scalar Multiplication Secure against Side Channel Attacks," Topics in Cryptology, The Cryptographers' Track at the RSA Conference 2003 (CT-RSA 2003), LNCS 2612, pp. 328{342,2003}.

[20] E. Al-Daoud, R, mahmod, Md. Rushdan, A. Kilicman (2002),“A new addition formula for Elliptic curve over GF (2n)”,IEEE Transactions on Computers, vol. 51, no. 8, pp. 972-975,Aug.