Australian Journal of Basic and Applied Sciences, 9(16) Special 2015, Pages: 473-482

ISSN:1991-8178

Australian Journal of Basic and Applied Sciences

Journal home page: www.ajbasweb.com

FPGA Based Architecture for Elliptic Curve Cryptography over GF (2^{256}) Using LFSR Data Selector and Influence of Multipliers in Resource Binding

[1]Jagan A and [2]Nagarajan V

[1]*Department of Computer Science and Engineering, Surya Group of Institutions, Vikiravandi, Villupuram, Tamil Nadu, India.*
[2]*Department Electronics and Communication Engineering, Adhiparasakthi Engineering College, Melmaruvathur, Kancheepuram, Tamil Nadu, India.*

ARTICLE INFO

Article history:
Received 12 March 2015
Accepted 28 April 2015
Available online 14 June 2015

Keywords:
Elliptic Curve Cryptography (ECC); Linear Feedback Shift Register (LFSR); Field Programmable Gate Array (FPGA).

ABSTRACT

Cryptography (ECC) is established as an encryption options for secured transmission of data in network. The focus of the researcher are on improving the throughput, security, speed and key length, and reducing the area. The efficiency of ECC is lies on the performance of scalar multiplication and hence the selection of adders and multipliers play an important role in ECC. The influence different multipliers can be investigated for better choices. This paper describes an ECC architectures over GF(2^{256}) using linear feedback shift register (LFSR) and three different multipliers, namely array multiplier, modified booth multiplier and hybrid low power encoded multiplier. These multipliers are used in the Montgomery scalar multiplication algorithm to perform point addition and point doubling and LFSR is used as a data selector. The synthesis result shows that the proposed ECC processor at 100 MHz frequency operation consumes power of 820mW, 760mW and 582mW and occupies 26909, 23700 and 14260 LUTs with array multiplier, modified booth multiplier and hybrid multiplier respectively. The architecture is implemented using spartan3E family device XC3S1600E using Modelsim 5.7 and Xilinx 9.2i.

To Cite This Article: Jagan A and Nagarajan V., FPGA Based Architecture for Elliptic Curve Cryptography over GF (2256) Using LFSR Data Selector and Influence of Multipliers in Resource Binding. ***Aust. J. Basic & Appl. Sci.,*** *9(16): 473-482, 2015*

INTRODUCTION

Information security and communication networks are interdisciplinary both in their technology and as well as their application domains (Yoo *et al.* 2014), (Suganthi and Sumathy 2014) and (Kishore Rajendiran *et al.* 2011). The ECC is independently proposed by Koblitz (Koblitz 1987) and Miller (Miller 1986), which has gained much popularity in industry and academia. The main success over the ECC is the high level of security can be achieved with shorter key, which is generated based on the strong mathematical operations such as discrete logarithms over integers or integer factorizations. The less computation time, less power consumption and less area occupy can be achieved with less key size compared with RSA. The most important operation in ECC is computing kP (a point or scalar multiplication), where k is an integer and P is a point on an elliptic curve. These operations can be computed by repeated point additions and doublings.

The ECC architecture over GF(p) and GF(2^m) have been proposed in (Satoh and Takano 2003). The 160-bit processor designed based on the Montgomery's scalar multiplication algorithm with projective coordinate and implemented in 0.13µm CMOS standard cell library. The high-speed design occupies 117.5 K-gates and operation times 0.19 ms for a 160-bit elliptic curve (EC) scalar multiplication in GF (2^m). A generic parallel architecture for fast EC scalar multiplication over binary fields has been demonstrated (Nazar *et al.* 2004). This 191-bit EC scalar multiplication operation is achieved in 56.44µS. Hardware implementations of elliptic and hyper-elliptic curve cryptography for binary fields using Digit-Serial Multiplier has been proposed in (Sandeep Kumar *et al.* 2006). An efficient flexible ECC architecture with novel word-level algorithms has been implemented using field-programmable gate-array (FPGA) (Mohammed Benaissa and Wei Ming Lim 2006). In (Kishore Rajendiran *et al.* 2008) a high performance 163 bit ECC processor has been proposed, which is based on a modified Lopez-Dahab elliptic curve point multiplication algorithm and uses Gaussian normal basis (GNB). This processor is designed using Xilinx XC4VLX80 FPGA device, which occupies 24,263 slices with

Corresponding Author: Jagan A, Department of Computer Science and Engineering, Surya Group of Institutions, Vikiravandi, Villupuram, Tamil Nadu, India.
E-mail: jagan.aa@gmail.com; Tel.: +91-7598469950.

maximum frequency of 143MHz. In (Chanho Lee and Jeongho Lee 2003) area trade-off architecture has been designed for a 193-bit finite field multiplier and an inversion unit is based on a normal basic representation. This processor is designed Verilog HDL and a 0.35μm CMOS cell library. In (Chanho Lee and Jeongho Lee 2003) a highly efficient general ECC architecture has been presented over GF(2^m), with the analysis the effects of parallelization. This architecture is implemented on an Altera Stratix II FPGA. In (Jarvinen and Skytta 2008) a flexible turbo decoding algorithm has been proposed for a high order modulation scheme that uses a standard half-rate turbo decoder designed for binary quadrature phase-shift keying (B/QPSK) modulation and its FPGA implementation consumed less latency and power compared with decoding.

The basic aim of this paper is providing a systematic approach for developing an ECC co-processor over binary field. The Montgomery's scalar multiplication algorithm has been used for 256-bit key length with LFSR based data selection. The other objective is evaluating the influence of different multipliers in performance and resource binding. The architecture is implemented using spartan3E family device XC3S1600E using Modelsim 5.7 and Xilinx 9.2i.

Elliptic Curve Cryptography:

The security of ECC is lies on the hardness of the mathematical computation which is based on the DLP (Itoh and Tsujii 1988), (Hankerson et al 2004). The discrete logarithm functions to the Abelian group formed by the points of an EC over a finite field. The IFP, the DLP, and the ECDLP are the three major computational problems visage on the applications of PKC. Fig. 1 provides a sample EC that is used to implement the cryptographic schemes. The elements of the group are the rational points on the EC, together with a special point O (called the "point at infinity").

Since ECC adapted to portable devices it has to occupy less area, low power and performed in high speed. One of the most time consuming operations in ECC is scalar multiplication (Schneier 1996), (Diffie and Hellman 1976), an operation of the form k.P, where, 'k' is a positive integer and 'P' is a point on the EC. Scalar multiplication k.P can be calculated by adding the point P to itself k-1 times and in addition the resulting point named as 'Q' should be on the EC. Another tedious operation on ECC is inverse operation i.e., to recover 'k' when the points 'P' and Q = k.P are given, is known as the ECDLP.

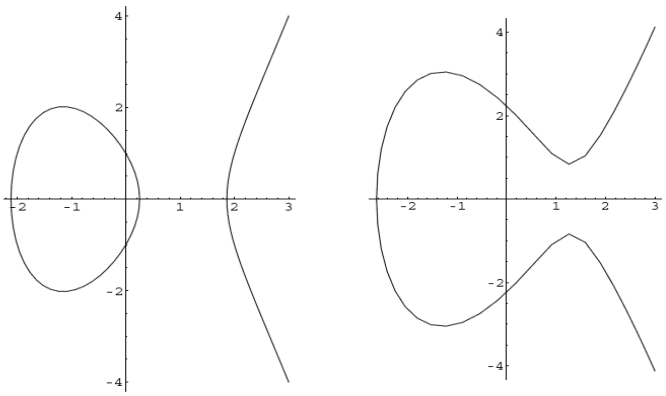

Fig. 1: Graphs of elliptic curves y2=x3-4x+1 (on the left) and y2=x3-5x+5 (on the right) over R

Proposed Architecture:

The proposed architecture is for the polynomial basis IEEE 1363 standard specifications for public key cryptography over the binary field GF(2^m), where the EC is defined as

$$y^2 + xy = x^3 + ax^2 + b \quad (1)$$

where a,b∈ GF(2^m) and b ≠ 0. If P is a point on ECE and k is a large integer, computation of the Q = kP , that is add P by k times, where Q is the point on the EC over GF(2^m).The operation kP can be performed by iterative point double and point addition.

The Fig. 2 shows the 256- bit binary ECC processor architecture for key generation based on projective coordinate using different multipliers namely array, modified Booth and hybrid multipliers. The Montgomery scalar multiplication algorithm gets input from the input buffer and generates the key for encryption which is given to output buffer. The architecture contains several modules they are LFSR, clock control, Montgomery scalar multiplication, point addition, point doubling, multipliers and clock control. The LFRS module select a point P(x,y) on EC curve also it does not allow repeated data and hence, it is very difficult for crypt analyzers using Brute-Force attacks; so it is highly secured.

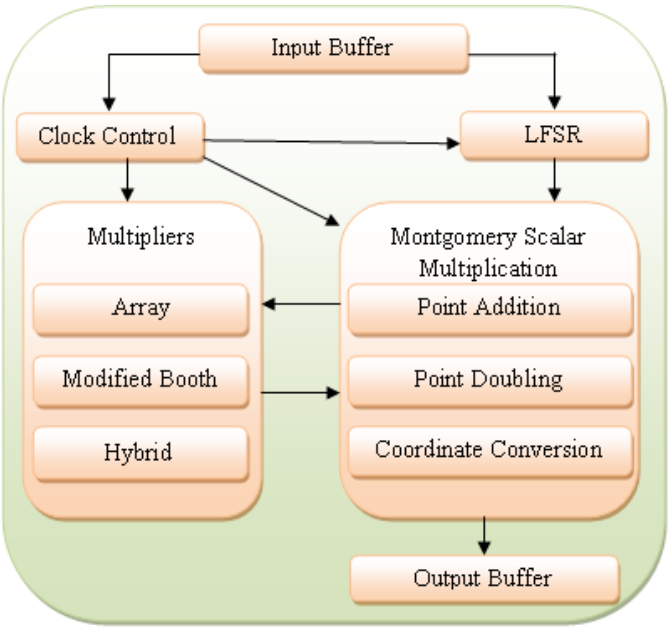

Fig. 2: Proposed ECC architecture with different multipliers

The output of LFSR module is given as input to Montgomery scalar multiplication module, which is generates the key for encryption by repeated addition and doubling. This module also performs the coordinate conversion. The point addition and point doubling operation are basic fundamental operation in multiplication process. The multiplier module enhances the performance of scalar multiplication. To performance the scalar multiplication three different multipliers are used and each multiplier is produced different outcome. The outcome of Montgomery scalar multiplication module is given to output buffer. Clock control unit issue the clock signal to all the modules.

Multipliers

4.1. Array Multiplier:

An array multiplier is commonly used to perform the multiplication among the two numbers in binary representation. It requires 'n' partial products for multiplying the two n-bit binary numbers. Array multiplier used in the applications such as general digital electronic devices, personal computers etc. Among two n-bit binary numbers, one number is multiplicand and the other is multiplier. The rules for performing the binary multiplication can be stated as follows.

(1) Check each and every bit in the multiplier.

(2) If the multiplier bit is 1, then multiplicand is simply copied down and represents the product line.

(3) If the multiplier digit is a '0', then product line is represent as '0'.

(4) The well designed array multiplier circuit inherits the following three capabilities.

- Identifying a bit as either '0' or '1'.
- Left shifting operation and
- Add all the partial products

The Fig. 3 describes that typical array multiplication requires eight partial products and seven additions to perform the 8-bit multiplication operation.

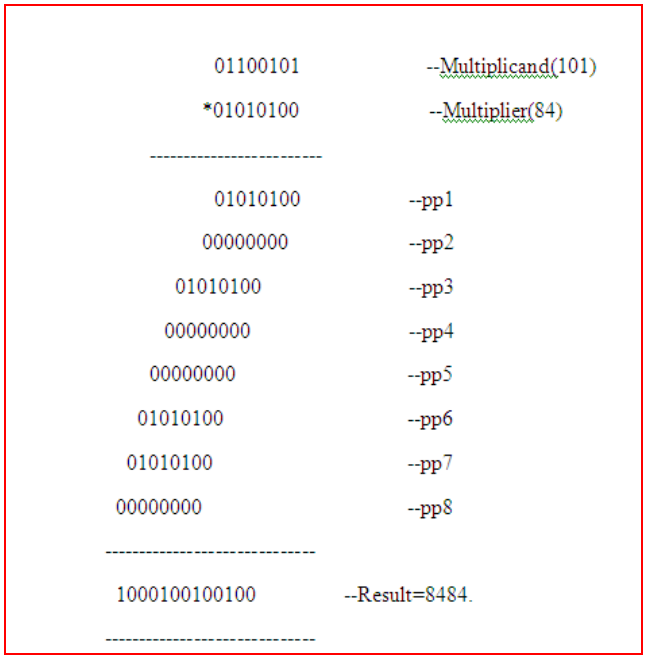

Fig. 3: Example of array multiplier

4.2. Modified Booth Multiplier:

In digital design another commonly used multiplier is modified Booth multiplier. It multiplies two binary numbers in two's complement representation and can handle signed bit numbers. The two important steps of modified Booth multiplication are shifting and addition. Generally this type of multiplication demands more number of shifting operations than additions. Shifting operations consume less power and are comparatively faster than addition. This algorithm basically does repeated addition of two predetermined values 'A' and 'S' to a product 'P' and then performs a rightward arithmetic shift on 'P' as illustrated in Table 1. A representative operation of modified Booth multiplier operation is shown in the Fig.4 with sample value of A=01100101 and B=01010100.

Table 1: Functionality of modified Booth multiplier

Xi+1	Xi	Xi-1	Action
0	0	0	0* Multiplicand
0	0	1	1* Multiplicand
0	1	0	1* Multiplicand
0	1	1	2* Multiplicand
1	0	0	-2* Multiplicand
1	0	1	-1* Multiplicand
1	1	0	-1* Multiplicand
1	1	1	0 (-0)* Multiplicand

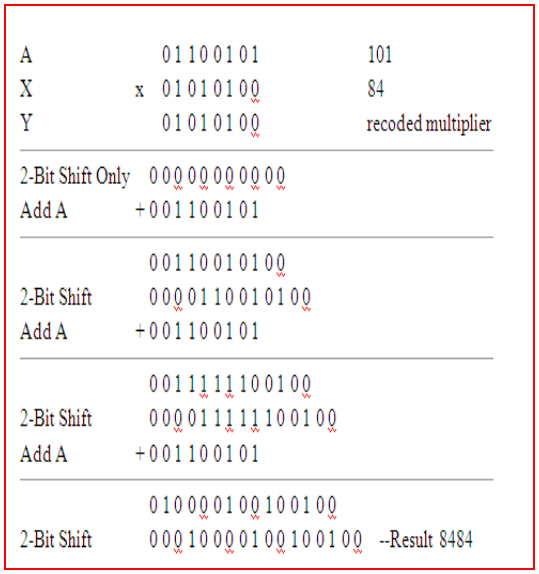

Fig. 4: Example - modified Booth multiplier

4.3. Hybrid Multiplier:

The efforts in the direction of performance enhancement of multipliers undoubtedly results in better ECC systems. The HELP multiplier is designed using Spurious Switching Suppression Technique (SSST). The number of bits in the input data plays vital role in the HELP multiplier. The dynamic power and switching activity of multiplication is mainly depending upon the input pattern in the given input. The principle of HELP multiplier is based on the number of 1's and its position in the input data.

The switching activity of HELP multiplier is reduced to 46 percentages compared to modified Booth multiplier. Due to the reduction of switching activity, the power of the HELP multiplier is reduced comfortably compared to the modified Booth multiplier. The operation of hybrid encoding rule is presented Table 2 (Saravanan and Madheswaran 2009 and 2010).

Table 2: Hybrid encoding scheme

No. of 1's in Multiplier	Position of bit '1'	Category	Operation
1	1^{st} bit	A	Add 0 to multiplicand (M)
1	i^{th} bit	B	Shift M left by i-1 and add 0
2	1^{st} and i^{th} bit	C	Shift M left by i-1 and add M
2	1^{st} and $i+j^{th}$ bit	D	Shift M left by j, add M and shift the result left by i-1
3	$i^{th}=1^{st}$, j^{th} and k^{th} bit	E	Shift M by k-j, add M and shift the result left by j-1,add M.
3	i^{th} , j^{th} and k^{th} bit	F	Shift M by k-j, add M and shift the result left by j-1,add M and shift the result by i.

According to category E, the HELP rule needs one partial product P1 with two additions. The remaining partial products P2 to P8 are zero, so the addition operation in this area can be neglected, which reduces the switching activity and power consumption. This spurious switching activity can be reduced by freezing the adders which perform this unwanted addition. A sample operation of hybrid multiplier is shown in the Fig. 5.

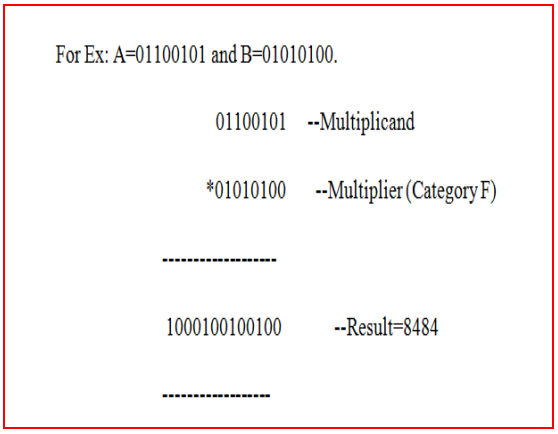

Fig. 5: Example of hybrid multiplier

RESULTS AND DISCUSSION

The proposed architecture is implemented using spartan3E family device XC3S1600E using Modelsim 5.7 and Xilinx 9.2i. Three different multipliers are used to enhance the performance of scalar multiplication. The synthesis results shows that architecture designed using the hybrid multiplier produce better result compared with other two architectures in terms of area, power and speed. The architecture designed using the hybrid multiplier consumed power with 582 mW and occupied 14260 LUTs in 100 MHz frequency compared with power consumption 760mW, area 23700 LUTs for modified Booth multiplier and power consumption 820mW, area 26909 LUTs for array multiplier.

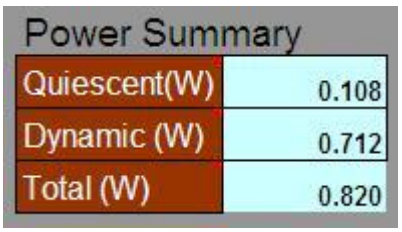

Fig. 6: Power summary- Array multiplier

Fig. 6. Shows the power summary of 256-bit ECC architecture with array multiplier, which includes both dynamic and static power consumed. The dynamic power is defined as amount of power consumed by switching activities of flip flops (FFs), where as static power is power consumed by leakage current. This architecture consumed 820mW with frequency 100MHz, which include both key generation and encryption operations.

Fig. 7: Power Summary - Modified Booth and Hybrid multiplier

Fig. 7 shows the power summary of 256-bit ECC architecture with modified Booth and hybrid multiplier. The architecture consumed 760mW and 582mW with frequency 100MHz for modified Booth and hybrid multiplier.

Synthesis results show that 256-bit ECC architecture with array multiplier occupy 26909 LUT's, with modified Booth multiplier occupy 23700 LUT's and with hybrid multiplier occupy 14260 LUT's. Among the three multipliers, the hybrid multiplier occupies less area. Fig. 8 shows the device utilization summary of ECC architecture with array multiplier, Fig. 9 shows, the device utilization summary of ECC architecture with modified Booth multiplier and Fig. 10 shows, the device utilization summary of ECC architecture with hybrid multiplier.

Device Utilization Summary		
Logic Utilization	**Used**	**Available**
Number of 4 input LUTs	26,906	29,504
Logic Distribution		
Number of occupied Slices	14,750	14,752
Number of Slices containing only related logic	14,695	14,750
Number of Slices containing unrelated logic	55	14,750
Total Number of 4 input LUTs	26,909	29,504
Number used as logic	26,906	
Number used as a route-thru	3	
Number of bonded IOBs	273	376
Total equivalent gate count for design	163,206	
Additional JTAG gate count for IOBs	13,104	

Fig. 8: Device utilization summary of ECC architecture with array multiplier

Device Utilization Summary		
Logic Utilization	**Used**	**Available**
Number of Slice Flip Flops	589	29,504
Number of 4 input LUTs	23,428	29,504
Logic Distribution		
Number of occupied Slices	13,713	14,752
Number of Slices containing only related logic	13,713	13,713
Number of Slices containing unrelated logic	0	13,713
Total Number of 4 input LUTs	23,700	29,504
Number used as logic	23,428	
Number used as a route-thru	272	
Number of bonded IOBs	274	376
IOB Flip Flops	170	
Number of GCLKs	1	24
Total equivalent gate count for design	203,772	
Additional JTAG gate count for IOBs	13,152	

Fig. 9: Device utilization summary of ECC architecture with modified Booth multiplier

Device Utilization Summary		
Logic Utilization	**Used**	**Available**
Total Number Slice Registers	174	29,504
Number used as Flip Flops	133	
Number used as Latches	41	
Number of 4 input LUTs	14,059	29,504
Logic Distribution		
Number of occupied Slices	7,813	14,752
Number of Slices containing only related logic	7,813	7,813
Number of Slices containing unrelated logic	0	7,813
Total Number of 4 input LUTs	14,260	29,504
Number used as logic	14,059	
Number used as a route-thru	201	
Number of bonded IOBs	274	376
IOB Flip Flops	170	
Number of GCLKs	2	24
Total equivalent gate count for design	119,520	
Additional JTAG gate count for IOBs	13,152	

Fig. 10: Device utilization summary of ECC architecture with hybrid multiplier.

The Fig. 11 shows, the sample encrypted cipher texts resulted of the proposed architecure using array multiplier, Fig. 12 shows, the sample encrypted cipher texts resulted of the proposed architecure using modifed Booth multiplier and Fig. 13 shows, the sample encrypted cipher texts resulted of the proposed architecure using hybrid multiplier.

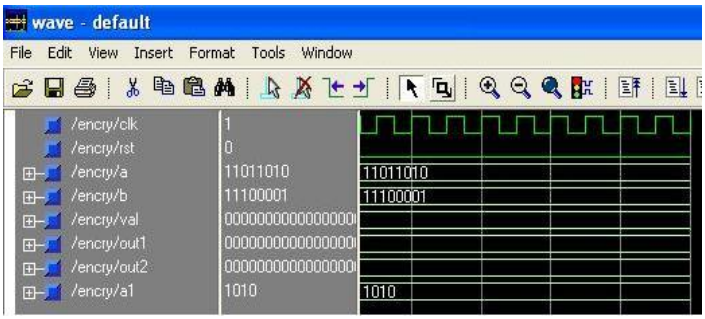

Fig. 11: Simulated encryption result of the proposed architecure using array multiplier

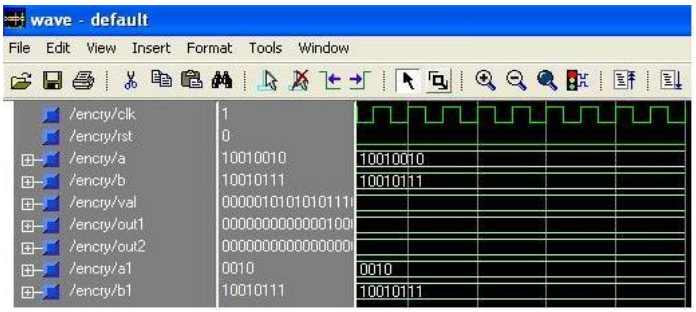

Fig. 12: Simulated encryption result of the proposed architecture using modifed Booth multiplier.

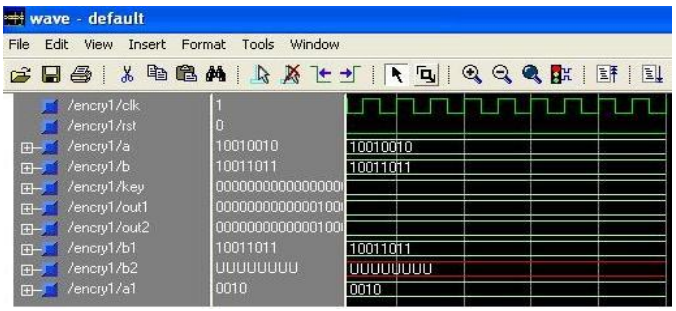

Fig. 13: Simulated encryption result of the proposed architecure using hybrid multiplier.

Related Works And Comparison:

The Table 3 describes the performance comparison of proposed 256-bit ECC processor with other similar design. Yaxun Gong and Shuguo Li (Yaxun Gong and Shuguo Li 2010) have proposed a high-throughput FPGA implementation of 256 –bit Montgomery Modular multiplier with 30.38MHz. The embedded 18x18 multiplier employed dedicates three cycles to computes a modular multiplication. The algorithm using Karatusba-Ofman multiplication reduced the number of multiplier while the five stage pipeline structure increased the throughput. Zhang et al (Zhang *et al.* 2009) have designed a 256-bit high speed ECC architecture using the Xilinx XC2V250 (Virtex-II) with 50MHz frequency. The architecture requires a high number of LUTs. The hardware architecture introduced by McIvor et. al. (McIvor *et al.* 2006) can perform the main prime field arithmetic functions including modular inversion and multiplication.

Table 3: Comparison of ECC Processor architectures

Design	Key Size	Technology	f(MHz)	Area	Arithmetic Unit
(Yaxun Gong and Shuguo Li 2010)	256	Altera Cyclone 3 EP3C40	30.38	---	81 embedded multipliers
(Zhang *et al.* 2009)	256	Xilinx XC2V 250(Virtex – II)	50	2594 LUTs	8 modular multiplication and 13 modular addition
(McIvor *et al.* 2006)	256	XC2VP125-7-ff1696	45.68	11,992 slices	Cascading numerous 16 × 16-bit unsigned multipliers

(Tanimura *et al*. 2008)	256	90nm CMOS	714	156 k gates	one radix-264 multiplier
(Ananyi *et al*. 2009)	256	Xilinx Virtex-4 XCV4FX100	60	20, 793 slices	embedded 18 × 18-bit multipliers
(Jagan and Nagarajan 2013)	256	XC3S1600E	100	27,114 LUT's	Array Multiplier
				23,693 LUT's	Modified Booth Multiplier
				13,938 LUT's	Hybrid Multiplier
Proposed works	256	XC3S1600E	100	26,909 LUT's	Array Multiplier
				23,700 LUT's	Modified Booth Multiplier
				14,260 LUT's	Hybrid Multiplier

The processor described uses a full-word multiplier which requires much fewer clock cycles than previous methods, while still maintaining a competitive critical path delay. A scalable unified dual-radix architecture for Montgomery multiplication in GF(P) and GF(2^n) has been proposed using CMOS technology (Tanimura *et al.* 2008). Ananyi et. al. has coined a flexible hardware processor for performing computationally expensive modular addition, subtraction, multiplication, and inversion over prime finite fields GF(p) (Ananyi *et al.* 2009). In (Jagan and Nagarajan 2013) the ECC co-processor over GF(2^{256}) is developed by using three different multipliers based on the Montgomery scalar multiplication algorithm. The multipliers, namely array multiplier, modified Booth multiplier and hybrid encoded low power multiplier are considered. The proposed 256- bit architecture is also designed using three different multipliers based on the Montgomery scalar multiplication algorithm. The performance of the proposed architecture is evaluated for three different multipliers at 100MHz by using LFSR as point identifier. The area utilization with hybrid multiplier is most economical.

Conclusion:

The ECC is widely used in portable device, which require low power, less memory and high performance during the communication. The performance enhancement is mainly depending on the using the better multiplier in the scalar multiplication operation. In this paper three various multipliers deployed namely array, modified Booth and hybrid low power encoded multiplier. The proposed 256-bit ECC processor can consumed power to perform the point multiplication and coordinates Conversion with 820mW, 760mW and 582mW with frequency 100 MHz for array, modified Booth and hybrid multiplier respectively. The area occupies for the proposed architectures are 26909 LUTs, 23700 LUTs and 14260 LUTs for array, modified Booth and hybrid multipliers respectively. The architecture is implemented using spartan3E family device XC3S1600E using Modelsim 5.7 and Xilinx 9.2i. Thus the paper explored the architecture possibilities to utilize different multipliers to speed up the computation of GF(2^m) elliptic curve crypto systems.

REFERENCES

Ananyi, K., H. Alrimeih and D. Rakhmatov, 2009. "Flexible Hardware Processor for Elliptic Curve Cryptography Over NIST Prime Fields", *IEEE Transaction on Very Large Scale Integration (VLSI) Systems*, 17(8): 1099 -1112.

Ansari and Hasan, M.A., 2008. "High-performance architecture of elliptic curve scalar multiplication", *IEEE Transaction on Computers*, 57(11): 1143-1153.

Chanho Lee and Jeongho Lee, 2003. "A Scalable Structure for a Multiplier and an Inversion Unit in GF(2^m)", *ETRI Journal*, 25(5): 315-320.

Duk Gun Choi et al., 2007. "An FPGA implementation of High-Speed Flexible 27-Mbps 8-State Turbo Decoder", *ETRI Journal*, 29(3): 363-370.

IEEE, IEEE 1363 Standard Specifications for Public-Key Cryptography, IEEE Standards Department, Piscataway, Jan. 2000.

Jagan, A. and V. Nagarajan, 2013. "A comprehensive performance investigation on ingenious ECC co-processor architecture for Different Multipliers", *Przeglad Elektrotechniczny*, R. 89 NR 8/2013, pp: 157-161.

Jarvinen, K. and J. Skytta, 2008. "On parallelization of high-speed processors for elliptic curve cryptography", *IEEE Trans. VLSI Systems*, 16(9): 1162-1175.

Kishore Rajendiran, Radha Sankararajan, and Ramasamy Palaniappan, 2008. "A Secure Key H. M. Choi, C. P. Hong, and C. H. Kim, "High performance elliptic curve cryptographic processor over GF(2163)", *in Proc. IEEE Int. Symposium on Electronic Design, Test, and Applications (DELTA), Hong Kong*, pp: 290-295.

Kishore Rajendiran, Radha Sankararajan, and Ramasamy Palaniappan, 2011. "A Secure Key Predistribution Scheme for WSN Using Elliptic Curve Cryptography", *ETRI Journal*, 33(5): 791-801.

Koblitz, N., 1987. "Elliptic curve cryptosystems," *Math. Comput.*, 48: 203-209.

Lai, J.Y. and C.T. Huang, 2008. "Elixir: High-throughput cost-effective dual-field processors and the design framework for elliptic curve cryptography", *IEEE Transaction on VLSI Systems*, 16(11): 1567-1580.

Lai, J.Y. and C.T. Huang, 2010. "High-Performance Architecture for Elliptic Curve Cryptography over Binary Field", *Proceeding of the IEEE International Symposium on Circuits and System (ISCAS)*, pp: 3933-3936.

McIvor, McLoone, and McCanny, 2006. "Hardware Elliptic Curve Cryptographic Processor Over GF(p)", *IEEE Transaction on Circuits and Systems I: Regular Papers*, 53(9): 1946-1957.

Miller, V., 1986. "Use of elliptic curves in cryptography," in Advances in Cryptology (CRYPTO), ser. Lecture Notes in Computer Science. New York: *Springer*, 218: 417–426.

Mohammed Benaissa and Wei Ming Lim, 2006. "Design of Flexible Elliptic Curve GF(2m) Cryptography Processors", *IEEE Transactions On Very Large Scale Integration (VLSI) Systems*, 14(6): 659-662.

Nazar A. Saqib, Francisco Rodrıguez-Henriquez and Arturo Dıaz-Perez, 2004. "A Parallel Architecture for Fast Computation of Elliptic Curve Scalar Multiplication over GF(2m)", *Proceeding of the18th IEEE International Parallel and Distributed Processing Symposium (IPDPS'04)*, pp: 144.

Sandeep Kumar, Thomas Wollinger, and Christof Paar, 2006. "Optimum Digit Serial GF(2m) Multipliers for Curve-Based Cryptography", *IEEE Transactions On Computers*, 55(10): 1306-1311.

Saravanan, S. and M. Madheswaran, 2009. "Design and Analysis of a Spurious Switching Suppression Technique Equipped Low Power Multiplier with Hybrid Encoding Scheme", *International Journal of Computer Science and Information Security*, 6(3): 73-78.

Saravanan, S. and Madheswaran, 2010. "Design of Hybrid Encoded Booth Multiplier with Reduced Switching Activity Technique and Low Power 0.13µm Adder for DSP Block in Wireless Sensor Node", *Proceeding on IEEE conference on wireless communication and sensor computing*, pp: 1-6.

Satoh and Takano, K., 2003. "A Scalable Dual-Field Elliptic Curve Cryptographic Processor," *IEEE Trans. Computers*, 52(4): 449-460.

Suganthi, N. and V. Sumathy, 2014. "Energy Efficient Key Management Scheme for Wireless Sensor Networks", *International Journal of computers communication & control*, 9(1): 71-78.

Tanimura, K., R. Nara and S. Kohara, 2008. "Scalable unified dual-radix architecture for Montgomery multiplication in GF(P) and GF(2n)", *Proceeding of the IEEE International Conference on Design Automation Conference*, pp: 697-702.

Yaxun Gong and Shuguo Li, 2010. "High-Throughput FPGA Implementation of 256-bit Montgomery Modular Multiplier", *Proceeding of the IEEE International Conference on Education Technology and Computer Science (ETCS)*, 3: 173-176.

Yoo, D., S. No and M. Ra, 2014. "A Practical Military Ontology Construction for the Intelligent Army Tactical Command Information System", *International Journal of computers communication & control*, 9(1): 93-100.

Zhang Jiahong, Xiong Tinggang, and Fang Xiangyan, 2009. "A Fast Hardware Implementation of Elliptic Curve Cryptography", *Proceeding of the IEEE International Conference on Information Science and Engineering (ICISE)*, pp: 1519–1522.