International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2022): 7.942

Implementation of Elliptic Curve Cryptography Processor for FPGA Applications

Ch.Venkateswarlu[1], Nirmala Teegala[2]

[1]Department of Electronics and Communication Engineering, CMR Engineering College, Hyderabad, Telangana State, India

[2]Department of Computer Science and Engineering, CMR Institute of Technology, Hyderabad, Telangana State, India

Abstract: ***ECC processor is implemented for point multiplication on FPGA based applications. High-precision segmented multiplier is used to reduce the latency and to avoid data dependency problem by modifying Lopez-Dahab Montgomery Point Multiplication algorithm.ECC processor using three multiplier, reduces the number of clock cycles required .In this paper ECC processor is implemented on Xilinx families' virtex-4 virtex-5 virtex-7.high performance can be achieved and number of clock cycles is reduced by using three multiplier ECC processor.***

Keywords: ECC (Elliptic Curve Cryptography), Latency, Galois field (GF), Clock Cycle (CC), Point Multiplication (PM), Multiplier (MUL)

1. Introduction to Elliptic Curves

In cryptography these elliptic curves are used for high security purpose, an elliptic curve is a plane curve, it may have discontinuous values defined as the $y^2 + xy = x^3 + ax^2 + b$.

An elliptic curve is a plane curve, it may have discontinuous values. It has so many properties which allow us the elliptic curves in cryptography.

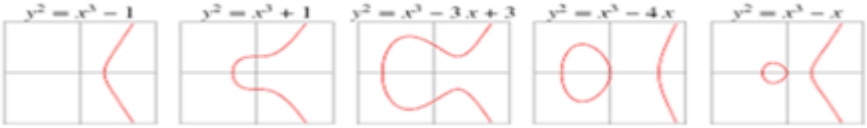

Figure 1: Elliptic Curves

1.1 Scalar Multiplication

The main operation in ECC is scalar pm, q = kp, where k is a private key, q is a public key, and p is a base point on an elliptic curve, e. the public key q is computed by k times point addition operation q = kp = p + • • • + p + p. The private k is difficult to retrieve from knowledge of q and p.

An elliptic curve over GF (2m) E can be defined as $y^2 + xy = x^3 + ax^2 + b$
Where a, b are constants, and a point at infinity is θ such that $P_{i+\theta} = P_1$, where $P_i = (x_1, y_1)$ and (x_1, y_1) GF (2^m).

The PM is achieved with scalar PM algorithms utilizing point addition and point doubling depending on the i^{th} value of K, K_i. Scalar PM can be affine coordinates based or projective coordinates based. Because of the expensive inversion operation involved in affine coordinates-based algorithms, projective coordinates-based PM is a more common choice for ECC hardware implementation. In this paper, the Lopez–Dahab (LD) Montgomery PM is considered. This algorithm requires six field multiplications, five field squaring operations, and four addition operations. The LD algorithm is generally faster to implement, and leads to improved parallelism and resistance to side channel power attack

1.2 Field Arithmetic GF

Field multiplication, field squaring, field addition, and field inversion operations are involved in a point operation. Addition and subtraction are equivalent over GF (2^m), which are very simple bitwise XOR operations. Field inversion is very costly in terms of hardware and delay. In projective coordinates, an inversion operation is used for the projective to affine coordinates' conversion that can be achieved with multiplicative inversion. The Itoh–Tsujii algorithm is selected as it requires only $\log_2(m)$ multiplications and (m−1) repeated squaring operations. In projective coordinates-based implementations, the overall performance depends on the performance of the field multipliers.

1.3 Elliptic Curve Properties

PM can be considered as the combination of PM addition and PM doubling.

If a line intersects two points on the curve then line intersects another point and its reflection gives original point.

- Line which is tangent to the curve intersects another point and its reflection at that point gives original point.

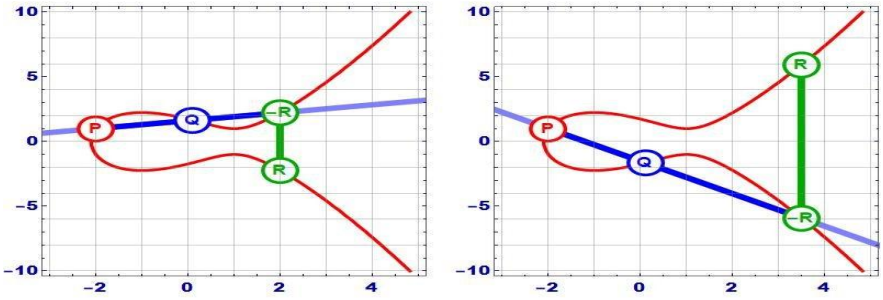

Figure 2: PM Addition

- Adding two points on the curve, P and Q are added to obtain P+Q which is a reflection of R along the X-axis.
- A tangent at P is extended to cut the curve at a point; its reflection is 2P

Adding P and 2P gives 3P and so on.

- Similarly, such operations can be performed as many times as desired to obtain Q = KP

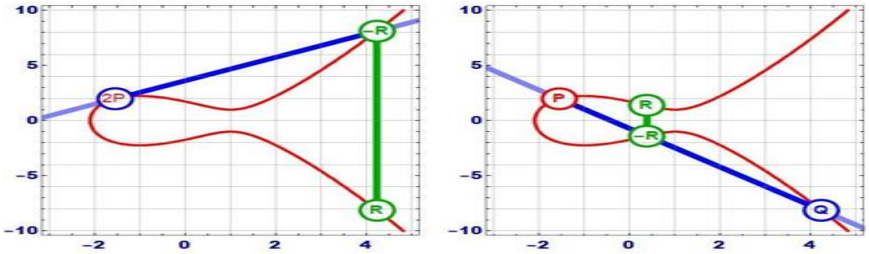

Figure 3: PM Doubling

Let us assume that p is the initial point and q is the final point

q=p1+p2+p3+p4+p5+p6+p7+p8+p9+…

q=p1+(p1+p1)+(p2+p1)+(p3+p1)+(p4+p1)+(p5+p1)+(p6+p1)+…

q=p1+(2p1)+(2p1+p1)+(2(2p1))+(2(2(p1))+p1)+(2(2p1))+2p1)+

To implement PM we need so many addition operations, squaring operations and inverse operations

For example

q=12p1

q=2(2p1) +2(2p1) +2(2p1)

To get q value, we need three additions, three multiplications and three doublings.

Table 1: Comparison with various technologies

Symmetric Encryption (Key size in bits)	RSA and Diffie- Hellman (modulus size in bits)	ECC Key size in bits
56	512	112
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	512

1.4 Elliptic Curve points

For a given elliptic curve we will find the curve points based on prime number or Galois field implementation. It is easy to calculate mod of prime values.

For example

1.5 ECC Advantages and Disadvantages

Equivalent ECC key size is 160 bits as compared to 1024 bit size of RSA.ECC does not require prime numbers and exponential processing for encryption. ECC offers considerable bandwidth savings when being used to transform short messages having very fast key generation .Moderately fast encryption and decryption ,it is widely used providing good protocols for authenticated key exchange.As binary curves are really fast in hardware,they can less storage and smaller chips are used with compact software.However ECC is mathematically more difficult to explain to client and complicated and tricky to implement securely.

2. Proposed High-Performance ECC Processor (HPECC)

ECC processor is implemented with high-precision m-multiplier with three two pipelining stages, one squaring circuit, one squad-squaring circuit, and two addition circuits to accomplish point operations i.e. point addition and point doubling in six CC's.

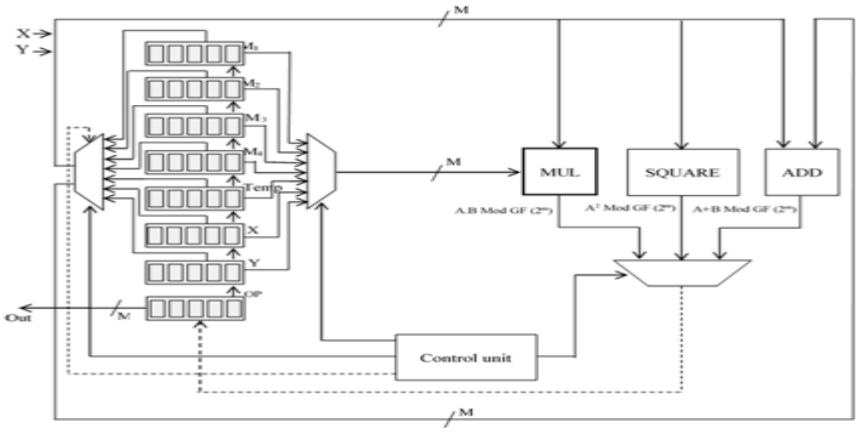

Figure 4: Hardware control architecture of control unit of ECC processor

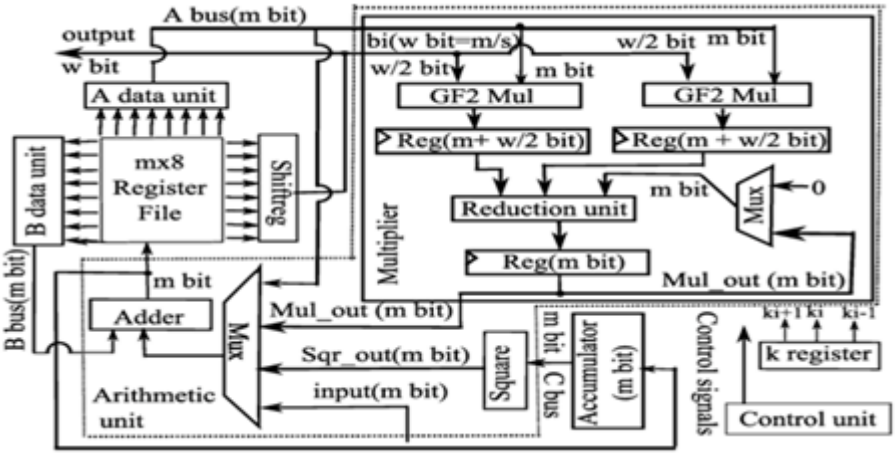

Figure 5: HPECC Processor Architecture

To avoid data dependency we combine point addition and point doubling. In PM method two stages of pipelining are overlapped with next loop .To obtain six clock cycle algorithm we will use square operations, double square operation and both operations in parallel as there is a data dependency problem as two pipeline stages are overlapping with next stage loop,

In our proposed architecture, we use register in the arithmetic data path to achieve a repeated quad-square operation without loading in to main memory. Proposed HPECC processor design uses a segmented pipelining – based full precision multiplier to achieve six CC for each loop of PM. Critical path delay of ECC processor depends on critical path delay of multiplier's and in turn multiplier critical path delay depend on path delay of GF^2MUL part or reduction part depending on the size of segment. Critical path delay of ECC can be combination of reduction part, adder, and multiplexer .main focus is on reduction of number of clock cycles .Our design can manage to take six CC's for each loop of PM

The total number of cc's for PM = 5 CC's (required for initialization) + 6*(m − 1) CC's (for PM in the projective coordinates) + CC's (for the final coordinates conversion = m/2 CC's for square + #MUL for inversion *3 + 3 CC's for inversion + 28 CC's for others) + 3 CC's for pipelining. The others clocks cycles that are independent of curve sizes are included: ten multiplications, six additions, and one square operation. For example, the total CC's for PM over GF with 163 bit = 5 + (6*162) + 139 (= (81 + 27 + 3) + 28) + 3 = 1119 cycles. Similarly, the latency of the HPECC processor over GF with 571-bit is 3783 CC's.

3. Proposed Low Latency ECC Processor (LLECC)

To achieve low latency high-speed ECC processor three full-precision multiplier are used such that to get six multiplications can be achieved in two steps for that ECC processor needs single-clocked field multipliers along with concurrent square and addition operations.

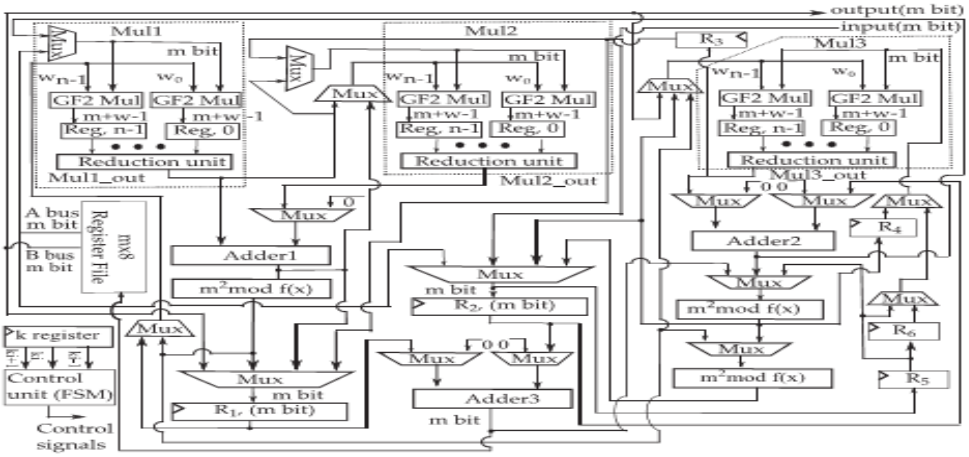

Figure 6: LLECC Processor Architecture

In modified PM three full–precision multipliers are used, in each state of proposed algorithm three multipliers output are concurrently used for addition ,square ,four-square to generate required output for the next states such to get three state multiplications in a single CC. To accomplish two CC's based operation, we need to process the multiplier output in the same CC by cascading the adder and squaring circuit.

The control unit of LLECC processor is also based on FSM that controls the two CC's based point operations and is simpler than control unit of HPECC processor. The critical path delay of the LLECC is the path delay of MUL GF 2+ the reduction part + adder + square + 3 * 1.The total number of CC's depends on the latency of loop operations of the PM.

The total number of CC's for PMs of the LLECC = 5 CC's for initialization +4 CC's to start of the loop + (m − 1)*2 CC's for loop operations +4 CC's to exit loop + CC's for coordinates' conversion [= (m/2) for square+ #mulx1) CC's for inversion +23 others]. The LLECC architecture consumes extra CC's at the start of first loop and at the end of the final loop operation due to load/unload of variables to/from the local registers. Again, the latency for inversion depends on the curve size and defined by $[\log_2 m - 1] + h(m - 1) - 1$, where h(m − 1) is the Hamming weight. The other CCs, 23 CC's that are independent of curve size mainly include ten multiplications, six additions, and one square operation. For example, the total number of CC's for GF of 163 bit = 5 + 4 + 162*2 + 4 + 113(= (81 + 9) + 23) = 450 CC's.

4. Results & Analysis

LLECC processor with three parallel multiplier increases the speed by decreasing the latency with little more area requirement.ECC processor with two pipeline stages with three parallel multipliers improves the speed by reducing the latency with little area overhead. ECC processor implementation with two stage pipelining to achieve high clock frequency achieves the fast is doubled and better area-time metric.

Proposed high-performance one-multiplier based architecture takes six cycles for a loop of Montgomery PM without pipelining delay, whereas our three-multiplier based processor takes only clock cycles. The proposed ECC processor is implemented on Xilinx FPGA families i.e. virtex-4, vitrtex-5, virtex-7 FPGA families resulted in fastest performance of the processor is obtained. On virtex-7 FPGA based processor implementation best area, time and fastest performance. Our parallel multiplier-based ECC design is the full-precision parallel architecture for the GF with 163-bit with lowest latency on FPGA environment.

Table 2: Comparison between various FPGA Technologies

Reference	Freq (MHZ)	Clock Cycles	FPGA	Resource MUX
HPECC_1M	210	1119	Vitex4	163bit
HPECC_1M	228	1119	Virtex5	163bit
LLECC_3M	113	450	Virtex5	3X163bit
HPECC_1M	352	1119	Virtex7	163bit
LLECC_3M	159	450	Virtex7	3X163bit
HPECC_1M	111	3783	Virtex7	571bit

References

[1] N. Koblitz, "Elliptic curve cryptosystems," Math. Comput. vol. 48 no. 177, pp 203–209, Jan. 1987.
[2] V. S. Miller, "Use of elliptic curves in cryptography," in Advances in Cryptology. Berlin, Germany: Springer, 1986, pp. 417–426.
[3] N. Koblitz, A. Menezes, and S. Vanstone, "The state of elliptic curve cryptography," Designs, Codes Cryptogr., vol. 19, nos. 2–3, pp. 173–193, Mar. 2000.
[4] D. Hankerson, A. J. Menezes, and S. Vanstone, Guide to Elliptic Curve Cryptography. New York, NY, USA: Springer-Verlag, 2004.
[5] National Institute of Standards and Technology (NIST), "Recommended elliptic curves for federal government use," Jul. 1999. [online]. Available http://csrc.nist.gov/encryption
[6] J. López and R. Dahab, "Fast multiplication on elliptic curves over GF(2m) without precomputation," in Proc. 1st Int. Workshop Cryptogr. Hardw. Embedded Syst., 1999, pp. 316–327.
[7] S. Kumar, T. Wollinger, and C. Paar, "Optimum digit serial GF(2m) multipliers for curve-based cryptography," IEEE Trans. Comput., vol. 55, no. 10, pp. 1306–1311, Oct. 2006.

[8] Z. U. A. Khan and M. Benaissa, "Low area ECC implementation on FPGA," in Proc. IEEE 20th Int. Conf. Electron., Circuits, Syst., Dec. 2013, pp. 581–584.
[9] T. Itoh and S. Tsujii, "A fast algorithm for computing multiplicative inverses in GF(2m) using normal bases," Inf. Comput., vol. 78, no. 3, pp. 171–177, Sep. 1988.
[10] B. Ansari and M. A. Hasan, "High-performance architecture of elliptic curve scalar multiplication," IEEE Trans. Comput., vol. 57, no. 11, pp. 1443–1453, Nov. 2008.
[11] S. S. Roy, C. Rebeiro, and D. Mukhopadhyay, "Theoretical modeling of elliptic curve scalar multiplier on LUT-based FPGAs for area and speed," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 21, no. 5, pp. 901–909, May 2013.
[12] W. N. Chelton and M. Benaissa, "Fast elliptic curve cryptography on FPGA, IEEE Trans, Very Large Scale Integr. (VLSI) Syst., vol. 16, no. 2, pp. 198–205, Feb 2008.
[13] G. D. Sutter, J.-P. Deschamps, and J. L. 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.
[14] Y. Zhang, D. Chen, Y. Choi, L. Chen, and S.-B. Ko, "A high performance ECC hardware implementation with instruction-level parallelism over GF(2163)," Microprocessors Microsyst., vol. 34, no. 6, pp. 228–236, Oct. 2010.

Author Profile

Ch.Venkateswarlu Received B-Tech in Electronics and Communication Engineering from RVR&JC college of engineering,Guntur.Completed ME in Microwave &Radar Engineering from Osmania university, Hyderabad. Currently working as Assistant Professor in CMR Engineering College, Hyderabad.

Nirmala Teegala Received B-Tech in Computer Science and Engineering from Sri Venkateshwara college of Engineering,Suryapet. M-Tech from Swarna Bharathi Institute of Technology and Science, khammam. Currently working as Assistant Professor in CMR institute of Technology, Hyderabad.