Parallel Processor Design for Binary Edwards and Huff Curves on FPGA with Latency Optimization

[a,] *J. Adline Vidhya , [b] V. R. Venkatasubramani, [c] S. Rajaram, [d] V. Vinoth thyagarajan,

[a,] Research Scholar, Department of Electronics and Communication Engineering , Thiagarajar College of Engineering, Madurai, India -625015,

[b,d] Associate Professor, Department of Electronics and Communication Engineering, Thiagarajar College of Engineering, Madurai,India-625015,

[c] Professor, Department of Electronics and Communication Engineering, Thiagarajar College of Engineering, Madurai,India-625015,

a*adline@student.tce.edu.

Abstract

Elliptic Curve Cryptography (ECC) is a powerful method for securing data, especially in devices with limited resources. This paper compares two efficient FPGA-based cryptographic processor designs—one using Binary Edwards Curves (BEC) and the other using Binary Huff Curves (BHC). Both designs aim to speed up the scalar multiplication operation, which is a key part of ECC.The BEC-based design improves performance by using multiple hybrid Karatsuba multipliers and a parallel version of the Hex Itoh–Tsujii algorithm for field inversion. This approach also reuses hardware between point operations and inversion, saving area and improving speed. It achieves 233-bit scalar multiplication in 0.033 ms on a Virtex-4 FPGA and 0.025 ms on a Virtex-7, showing 13% and 17% latency improvements compared to earlier designsThe BHC-based design benefits from a unified structure that performs point addition and doubling in a similar way, which helps resist power-based side-channel attacks. It uses the Non-Adjacent Form (NAF) method for scalar multiplication to further reduce power use. With parallel field arithmetic and optimized hardware blocks, it achieves low latencies of 29.5 ns (Virtex-4) and 23.1 ns (Virtex-7), reducing clock cycles by up to 59.6% and 42.3%, respectively.Overall, the BEC processor offers better throughput and efficient use of hardware, while the BHC processor provides strong security and fast computation. This comparison highlights the trade-offs between speed, area, and security in FPGA-based ECC designs.

Keywords— Finite-Field Inversion (FF-Inversion),Elliptic Curve Cryptography (ECC) Itoh-Tsujii Algorithm (ITA) Hex Itoh-Tsujii Algorithm (H-ITA),NAF (Non-Adjacent Form), BEC(Binary Edward Curve),BHC (Binary Huff Curve).

1. INTRODUCTION

Elliptic Curve Cryptography (ECC), introduced independently by Miller [2] and Koblitz [1], is widely used in secure communication systems due to its ability to offer strong security with smaller key sizes compared to traditional cryptosystems like RSA. This makes ECC an excellent choice for resource-constrained environments such as IoT devices and embedded systems [3], [4]. ECC can be implemented over prime fields or binary fields, with the latter being particularly efficient for hardware-based systems. However, traditional ECC operations—point addition and point doubling—use different formulas, which results in irregular execution patterns and exposes implementations to side-channel attacks (SCAs), such as Simple Power Analysis (SPA) [6], [7]. Moreover, ECC's incomplete formulas often require special case handling, such as when points at infinity are involved, making implementations more complex and error-prone [8].To address these issues, unified point addition formulas have been proposed to use the same logic for both point addition and doubling, thereby enhancing regularity and side-channel resistance [4]. Two popular models that support unified operations are Binary Huff Curves (BHC) and Binary Edwards Curves (BEC). BHCs are known for their uniform execution paths, which reduce power-based leakage. Recent designs have adopted the Non-Adjacent Form (NAF) for scalar multiplication and incorporated parallel hybrid Karatsuba multipliers along with the Hex Itoh–Tsujii algorithm for efficient field inversion. These optimizations achieved latencies of 0.029 ms and 0.023 ms on Xilinx Virtex-4 and Virtex-7 FPGAs, respectively [4], [5].On the other hand, BECs, introduced by Bernstein et al. [9], resolve ECC's completeness issues and offer complete, unified point operations without special-case handling. Their algebraic structure enables highly parallel hardware implementations. Using parallelized scalar multiplication techniques, hybrid Karatsuba multipliers, and a modified Hex Itoh–Tsujii algorithm, BEC-based architectures achieved 233-bit scalar multiplication in 0.033 ms (Virtex-4) and 0.025 ms (Virtex-7), showing significant latency improvements over earlier designs [10]. In this comparative study, we analyze and evaluate the FPGA-based implementations of both BHC and BEC over GF(2^{233}), focusing on latency, hardware resource usage, and resistance to side-channel threats. While BEC implementations offer better throughput and resource reuse, BHC designs excel in security-critical applications due to their consistent execution and reduced power leakage. This study helps guide the selection of appropriate curve models based on application-specific requirements in terms of performance, area, and security.

II. PRELIMNARIES

A. Binary Edwards Curves (BECs)

Binary Edwards Curves (BECs), designed for binary fields GF(2^m) where m≥3, are a class of elliptic curves with properties favorable for cryptographic applications due to their completeness and unified addition laws [9]. The general affine form of a BEC is given by:

$$d_1(X+Y)Z^3 + d_2(X^2+Y^2)Z^2 = XYZ^2 + XY(X+Y)Z + X^2Y^2 \tag{1}$$

Here, the constants d1,d2∈GF(2^m) , with d1≠0 and d1≠d2. A key advantage of BECs is the efficient computation of the negation of a point, which has negligible overhead and results in a point of order two. Additionally, the curve supports complete addition laws that allow all point pairs to be processed through the same formula without exceptions [9]This unified approach ensures the same formula can be used for both point addition and point doubling. As a result, there is no observable difference in power usage, enhancing protection against side-channel attacks like Simple Power Analysis (SPA) [11]. Furthermore, BECs exhibit a symmetry such that if a point (x_1,y_1) lies on the curve, then (y_1,x_1) also lies on the same curve [11].To reduce the computational cost of field inversion in affine coordinates, BECs are typically implemented in **projective coordinates** using the **Lopez–Dahab (LD) representation**. In this format, an affine point (x,y) corresponds to a projective point (X:Y:Z), where: x=X/Z,y=Y/Z^2 . The unified point addition formula for projective BECs with d1=d2 is expressed using the following intermediate variables [9]:

Table 1. Projective Addition Points for BEC Curve.

$A=X_1.X_2$
$B= Y_1.Y_2$
$C= Z_1.Z_2$
$D=d_1.C$
$E=C^2$
$F=d_1^{2}.E$
$G=(X_1+Z_1). (X_2+Z_2)$
$H=(Y_1+Z_1). (Y_2+Z_2)$
I=A+G
J=B+H
$K= (X_1+Y_1). (X_2+Y_2)$
$L=d_1.K$
U=C.(F+L.(K+I+J+C))
$V=U+D.F+L.(d_1.E+G.H+A.B)$

X_3=V+D.(A+D).(G+D)
Y_3= V+D.(B+D).(H+D)
Z_3=U

This formulation requires approximately **17 field multiplications**, offering constant-time operation and robustness against SPA [11].

B. Binary Huff Curves (BHCs)

Binary Huff Curves (BHCs) are a lesser-known family of elliptic curves originally proposed in classical number theory by Huff and later extended to binary fields. While early work on Huff curves focused on fields of odd characteristic [5], Devigne and Joye introduced a formal definition for binary fields in 2011 [12].

The general projective form of a BHC over GF(2^m) is defined by:

$$E_{F_{2^m}}: aX(Y^2 + YZ + Z^2) = bY(X^2 + XZ + Z^2) \qquad (2)$$

where $a, b \in F_{2^m}$ and a ≠ b. There are three points satisfying the curve equation, namely (a :b : 0), (1 : 0 : 0) and (0 : 1 : 0). The affine form of the binary Huff curve, corresponding to "(1)", is expressed as:

$$ax(y^2 + y + 1) = by(x^2 + x + 1) \qquad (3)$$

This equation is birationally equivalent to a Weierstrass-form elliptic curve, enabling useful transformations between different curve models for applications like digital signatures or key exchange [13]. The original unified addition formula by Devigne and Joye was designed to use a single computation path for both point addition and doubling, which provides inherent resistance to SPA [12]. However, in 2013, Ghosh et al. discovered that this approach could still leak information during doubling operations due to zero intermediate values. To overcome this, they proposed an updated unified addition formula with consistent intermediate computations, improving SPA resistance [14]. In projective coordinates, the improved point addition for BHCs involves:

Table 2. Projective Addition Points for BHC Curve

$m_1 = X_1X_2$
$m_2 = Y_1Y_2,$
$m_3 = Z_1Z_2$
$m_4 = (X_1 + Z_1)(X_2 + Z_2) + m_1 + m_3$
$m_5 = (Y_1 + Z_1)(Y_2 + Z_2) + m_2 + m_3$
$m_6 = m_1m_3$

$m_7 = m_2 m_3$
$m_8 = m_1 m_2 + m_3^2$
$m_9 = m_6(m_2 + m_3)^2$
$m_{10} = m_7(m_1 + m_3)^2,$
$m_{11} = m_8(m_2 + m_3)$
$m_{12} = m_8(m_1 + m_3)$
$X_3 = m_4 m_{11} + \beta \cdot m_9$
$Y_3 = m_5 m_{12} + \gamma \cdot m_{10}$
$Z_3 = m_{11}(m_1 + m_3)$

with constants $\beta = \frac{a+b}{b}$ and $\gamma = \frac{a+b}{a}$ This formulation also uses approximately **17 field multiplications**, aligning it with BEC's computational cost and maintaining constant operation time across all inputs [14].

III. FPGA Based ECC Processor Architecture

This section presents a unified FPGA architecture for elliptic curve cryptographic (ECC) processors implemented over binary curves, focusing on scalar multiplication, which is the most computationally intensive operation in ECC. The design facilitates a comparative evaluation of two prominent binary curve models—Binary Edwards Curves (BECs) and Binary Huff Curves (BHCs)—implemented within a common hardware framework. The processor architecture integrates optimized scalar handling, register management, and arithmetic operations tailored to support both curve types efficiently while highlighting their performance trade-offs in speed, area, and side-channel resistance [14], [15].

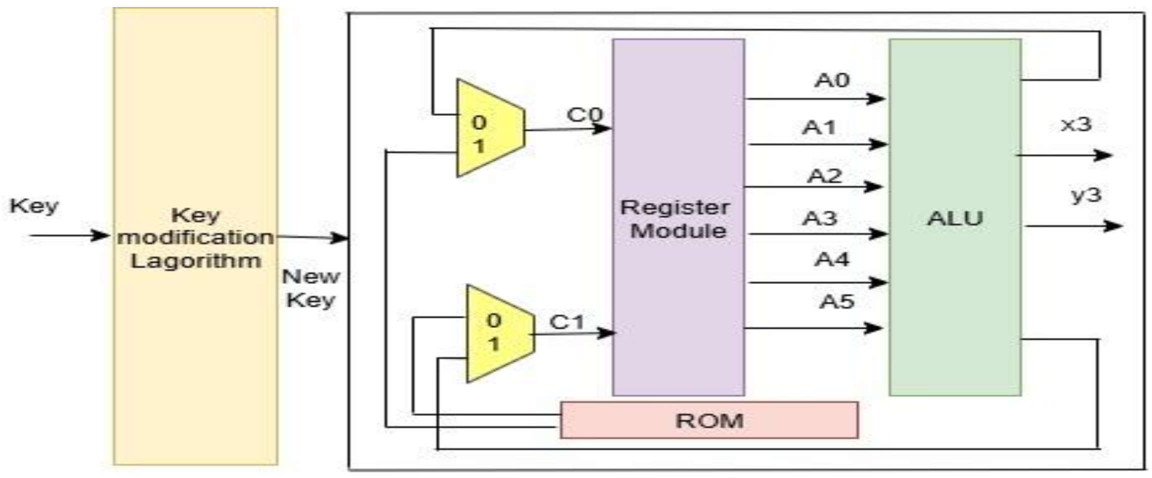

Fig. 1. Proposed FPGA Based ECC Processor Architecture

A. Architectural Overview

The processor is composed of three major modules: a Key Modification Unit for transforming scalars into compact forms, a Register Module for storing intermediate variables and constants, and an Arithmetic and Logic Unit (ALU) for executing finite field operations. The architecture incorporates two hybrid Karatsuba multipliers that support parallel field multiplication, leading to reduced clock cycles during point computations [16]. Inversion is carried out using a modified Hex Itoh–Tsujii Algorithm, enabling efficient projective-to-affine coordinate conversion [17]. To maintain regular computation flows and suppress side-channel leakage, the architecture adopts a unified control logic that supports both BEC and BHC without requiring structural changes. This flexibility permits direct performance comparisons between the curves under identical hardware and control conditions. The processor achieves scalar multiplication for 233-bit keys in just 0.033 ms and 0.025 ms on Virtex-4 and Virtex-7 FPGAs, respectively, demonstrating high throughput and low latency for both curve models [18].

B. Scalar Representation and Key Transformation

The architecture employs Non-Adjacent Form (NAF) representation for scalar values to reduce the number of point additions and overall computation time [19]. Scalar transformation proceeds by analyzing each bit of the binary scalar; if the value is odd, a non-zero digit is computed as 2 minus the scalar modulo 4, then subtracted from the scalar. For even values, a zero is appended, and the scalar is halved. The result is a sparse sequence composed of $\{-1, 0, 1\}$ elements. Scalar multiplication is then performed by iterating through the NAF digits, executing point doubling followed by conditional addition or subtraction based on the digit value. This technique reduces the effective number of point additions by nearly two-thirds [20]. Further optimization is introduced by encoding the scalar using ternary values with two-bit symbols, enabling simultaneous point computations and key processing [21]. The key modification logic is tuned such that for a 233-bit scalar, only 14 initial clock pulses are required before multiplication begins, in contrast to traditional designs that require up to 233 cycles for key loading. This optimization remains consistent across both BEC and BHC implementations, allowing meaningful architectural comparisons in terms of delay and resource usage.

C. Register Module Design

The processor includes ten m-bit registers for storing operands, intermediate results, and curve constants. To simplify the hardware design, both BEC and BHC implementations use equal constant parameters where applicable, facilitating a unified addition law that reduces register complexity [22]. The register module uses two input multiplexers (MC0 and MC1) and three output multiplexers to manage operand selection and routing to the ALU. A 24-bit control word governs register operations and multiplexer configurations, ensuring flexible and coordinated data flow for various scalar multiplication steps. The modular design allows the same register framework to support both curves without requiring architecture-specific modifications, making it possible to evaluate the trade-offs in resource utilization and execution time [23].

D. Arithmetic and Logic Unit

The ALU performs field addition using XOR gates and multiplication using hybrid Karatsuba logic optimized for FPGA implementations [24]. The availability of two multipliers allows concurrent evaluation of independent partial products during point addition and doubling. Field inversion, a traditionally expensive operation, is accelerated using a modified version of the Hex Itoh–Tsujii Algorithm, which exploits efficient squaring and multiplication sequences based on a precomputed addition chain [17], [25]. The ALU supports consistent execution timing by using regular control signals and minimizing data-dependent behavior, enhancing resistance to timing-based side-channel attacks [26]. The hardware-friendly implementation of field operations allows the processor to operate seamlessly for both curve models while maintaining high throughput and minimal latency.

E. Comparative Analysis of Curve Implementations

The architecture enables a direct performance comparison between Binary Edwards and Binary Huff Curves within a shared hardware framework. Both curves benefit from the same ALU, register structure, and key transformation module, allowing their computational characteristics to be evaluated under identical FPGA conditions. In practice, the Huff curve shows slightly lower register usage and improved doubling efficiency, while the Edwards curve offers stronger resistance to exceptional cases in point addition [27]. Despite minor algebraic differences, the unified processor design supports both with negligible reconfiguration. Experimentally, the processor achieves scalar multiplication for 233-bit keys in 0.033 ms and 0.025 ms for Virtex-4 and Virtex-7 FPGAs, respectively, across both curve models [18]. These results highlight the architectural balance achieved by the design and validate its suitability for flexible, high-performance elliptic curve cryptography on reconfigurable hardware platforms.

IV. Parallelism in Unified Addition

Parallelism in unified addition plays a pivotal role in minimizing the latency associated with scalar multiplication in elliptic curve cryptographic systems. In our comparative analysis of Binary Huff and Binary Edwards Curves, we explored how parallel multipliers contribute to latency reduction and improved computational efficiency.For the Binary Huff Curve, employing two parallel multipliers in the unified addition stage significantly optimizes data dependencies, allowing both multipliers to operate concurrently. This configuration leads to high utilization and enhances throughput performance. As observed in Table I, this parallelism reduces the total number of computational steps from 20 to 12, resulting in a latency reduction of one delay unit (1D). Although the architecture requires an additional multiplier unit (1M), the trade-off favors latency reduction and supports low-latency, high-throughput cryptographic applications. The architecture's parallel design also minimizes dependency chains, allowing for more efficient execution cycles, which is particularly beneficial in scenarios where rapid response is essential [29].In the case of the Binary Edwards Curve, a similar strategy of using two parallel multipliers is applied to the unified addition operation. Assuming SSS, MMM, and DDD represent the respective latencies for squaring, field multiplication, and constant multiplication, the latency for unified addition using a single multiplier is 21M + 1S + 4D [30]. However, by introducing a second multiplier, the latency is reduced to 16M + 2D, achieving a utilization factor of approximately 70%. Further optimization enables the reduction of delay to 16M + D, indicating that adding one multiplier eliminates the need for one constant multiplication. Notably, employing more than two multipliers offers no additional latency benefits. Furthermore, the architecture incorporates parallel addition storage to reduce the number of iterations, particularly evident in step-6 where intermediate values (RA2, A2, D1, B2, C2, E2, and E1) are processed in parallel. This architectural choice reduces the number of processing steps from 15 to 13, balancing area overhead with performance gains [31]. Overall, parallelism in unified addition enhances both Binary Huff and Binary Edwards Curve implementations, enabling faster scalar multiplication and supporting efficient cryptographic processing in FPGA-based systems. The comparative results highlight that while both curves benefit from dual multiplier setups, the internal architectural strategies and resource utilizations vary, offering flexibility for design trade-offs depending on the target application.

Table 3 : Data dependency of unified point addition in BHC and BEC

Step	**Operation C1**		**Operation C0**	
	BEC	**BHC**	**BEC**	**BHC**
1	$RA_2 \leftarrow X_2$	$RA_1 \leftarrow X_1$	$RB_2 \leftarrow Y_2$	$RA_2 \leftarrow X_2$
2	$RC_1 \leftarrow Z_1$	$RB_1 \leftarrow Y_1$	$RC_2 \leftarrow Z_2$	$RB_2 \leftarrow Y_2$
3	$RA_1 \leftarrow X_1$	$RC_1 \leftarrow Z_1$	$RB_1 \leftarrow Y_1$	$RC_2 \leftarrow Z_2$
4	$RE_1 \leftarrow A_2.A_1$	$RD_1 \leftarrow C_1.C_2$	$RD_1 \leftarrow B_1.B_2$	$RD_1 \leftarrow (A_1+C_1).(A_2+C_2)$
5	$RE_2 \leftarrow C_1.C_2$	$RE_1 \leftarrow A_1.A_2$	$RA_2 \leftarrow (A_1+B_1).(A_2+B_2)$	$RE_2 \leftarrow (B_1+C_1).(B_2+C_2)$
6	$RB_2 \leftarrow (B_1+C_1).(B_2+C_2)$	$RA_1 \leftarrow B_1.B_2$	$RD_2 \leftarrow (A_1+C_1).(A_2+C_2)$	$RA_2 \leftarrow (D_1.E_1)$
7	$RC_1 \leftarrow d.A_2$	$RB_1 \leftarrow A_1.D_1$	$RD_2 \leftarrow D_2.E_2$	$RB_2 \leftarrow E_1..A_1 + (D_1)^2$
8	$RB_1 \leftarrow (B_2+D_2).(D_1+D_2)$	$RC_1 \leftarrow (A_1+D_1)$	$RA_1 \leftarrow (E_1+D_1).(C_2+A_2)$	$RC_2 \leftarrow (E_1+D_1)$

9	$RA_2 \leftarrow (D_2)^2+C_1.A_2$	$RD_1 \leftarrow A_{2.}\ (C_1)^2$	$RE_1 \leftarrow C_{2.}\ B_2$	$RA_1 \leftarrow B_{1.}\ (C_2)^2$
10	$RD_1 \leftarrow E_1.\ D_1$	$RA_2 \leftarrow B_2.\ C_1$	$RD_1 \leftarrow d_{1.}\ (E_2)^2 +D_1+E_1$	$RB_1 \leftarrow B_2.C_2$
11	$RC_2 \leftarrow A_2.E_2$	$RC_2 \leftarrow A_2.C_2$	$RD_1 \leftarrow C_2+C_1\ .D_1$	$RD_1 \leftarrow \alpha D_1+(E_2+A_2).\ A_2 +(A_2)^2 + C_1$
12	$RD_1 \leftarrow D_1+D_{2.}\ .\ (D_2)^2$	$RB_2 \leftarrow \beta A_1+(E_2+B_1).\ B_1 +(B_1)^2 + C_1$	$RC_1 \leftarrow Z_1$	--
13.	$RA_2 \leftarrow D_1+D_{2.}A_1$		$RB_2 \leftarrow D_1+D_{2.}B_1$	

V. ALU Of The Binary Edward Curve and Huff Processor

The Arithmetic Logic Unit (ALU) is designed to support both Binary Edwards Curve (BEC) and Binary Huff Curve (BHC) operations using two parallel field multipliers to speed up scalar multiplication. This parallelism helps reduce clock cycles and improves performance. In the BHC processor, the use of two multipliers lowers the unified addition steps from 20 to 12, as shown in Table I. Although this adds one extra multiplier, it reduces delay due to fewer constant multiplications, improving speed and area efficiency [32]. For the BEC processor, the dual multiplier setup reduces the number of operations from 21 multiplications + 1 squaring + 4 constants to 16 multiplications + 2 constants. This cuts down the steps from 15 to 13 (Table II). About 70% multiplier utilization is achieved, and step-6 is optimized with parallel addition storage [33]. The ALU is controlled by input multiplexers (MC0, MC1) and output multiplexers (MuxOUT1–3), ensuring fast and smooth data flow. An optimized FSM further reduces idle cycles by allowing some operations to run in parallel [34], [35].The design achieves 0.038 ms on Virtex-4 and 0.031 ms on Virtex-7, showing up to 17% improvement over earlier designs [36]. This unified ALU supports both curve types efficiently, offering flexibility and speed for secure, real-time systems like 5G and intelligent transport [32]–[36].

VI. Implementation Results and Discussion

The proposed Binary Edwards Curve (BEC) and Binary Huff Curve (BHC) architectures were developed using Verilog HDL and implemented on Xilinx Virtex-7 FPGAs using the ISE simulator. The primary goal of both designs was to minimize **latency** rather than area, by focusing on reducing the number of clock cycles required for scalar multiplication. This operation includes both point addition (PA) and point doubling (PD), which are critical for cryptographic computations. In both architectures, scalar multiplication was performed for a field size of 233. The BEC-based processor required **3 clock cycles for initialization**, **10 clock cycles per key bit**, and **13 cycles for inversion**, totaling **2336 clock cycles**. In contrast, the BHC-based design optimized unified addition to require only **9 clock cycles per bit**, reducing the total to **2104 clock cycles**. This reduction is primarily due to efficient scheduling and parallel execution of operations using dual multipliers [37]. Latency, calculated using the formula Latency, calculated as $T = \frac{ClockCycles}{f}$improved in both designs despite slightly reduced clock frequencies. The BEC-based design achieved a **233-bit point multiplication** latency of **0.033 ms** on Virtex-4 and **0.025 ms** on Virtex-7, improving upon previous designs by **13% and 17%**, respectively [38]. Similarly, the BHC-based processor achieved latency figures of **29.5 ns** on Virtex-4 and **23.1 ns** on Virtex-7. This translates to latency reductions of **59.6% and 42.3%**, respectively, compared to earlier implementations, and further improvements of over **20%** compared to the most

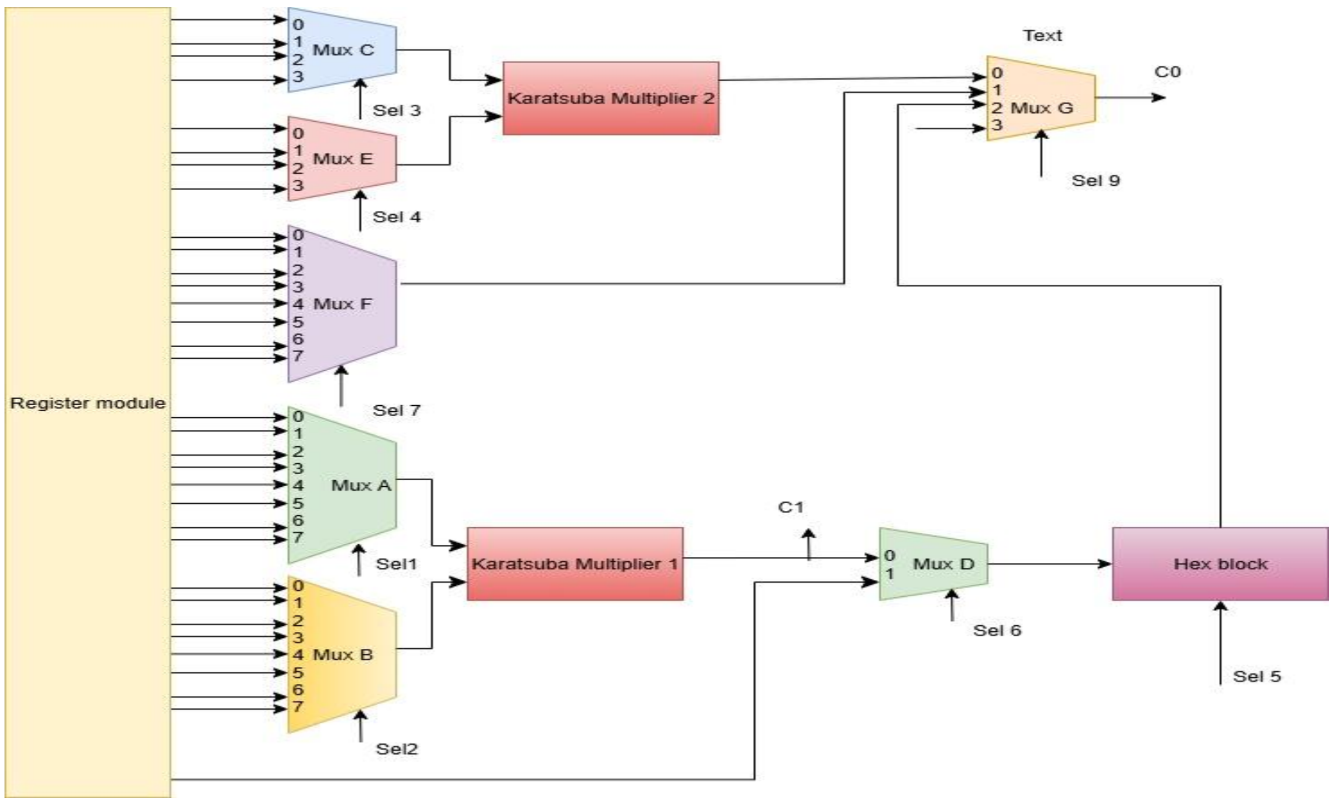

FiG. 2. ALU of FPGA Based ECC Processor

recent FPGA-based ECC processors [39]. The key improvement in both processors stems from the use of **parallel multipliers** and optimized finite field arithmetic operations, including Hex and Quad-based Itoh–Tsujii

Table-4: Performance Comparison of BEC and BHC Cryptoprocessors (233-bit Field)

Curve Type	FPGA Platform	Key Length (bits)	Slices	LUTs	Gate Count	Frequency (MHz)	Latency (ms)	Latency/bit (ns)
BEC	Virtex-4	233	33,504	64,058	479,766	70.216	0.033	141
BEC	Virtex-7	233	–	65,035	–	90.1	0.025	107
BHC	Virtex-4	233	–	–	–	71.3	0.0295	126
BHC	Virtex-7	233	–	–	–	90.9	0.0231	99

inversion strategies. While this introduces a marginal increase in the number of logic units (LUTs), the trade-off results in significantly lower delay. Additionally, both designs benefit from reduced register usage—approximately **2m fewer slice registers**—which is valuable given the large field sizes typical in ECC-based cryptographic applications [40]. Table 3 summarizes and compares the performance of both BEC and BHC processors with existing works. The BEC architecture, with its unified structure for point operations, offers balanced resource usage and reduced latency. The BHC processor further enhances performance by lowering the unified addition latency and achieving better throughput using tightly scheduled parallel operations [41].

VII. CONCLUSION

This work presents a comparative evaluation of FPGA-based cryptographic processors for Binary Huff Curve (BHC) and Binary Edwards Curve (BEC), with emphasis on latency, resource efficiency, and security. Both architectures adopt parallel multipliers and unified addition logic to enhance the efficiency of 233-bit scalar multiplication. The BHC implementation achieves 2104 clock cycles by optimizing point addition and inversion operations while minimizing register overhead, resulting in a compact area-time product. This makes it particularly suitable for resource-constrained applications, such as secure 5G systems, where side-channel resistance is also critical.In comparison, the BEC processor completes scalar multiplication in 2336 clock cycles, leveraging complete unified addition and inversion strategies to ensure both speed and side-channel protection. Despite a slightly higher cycle count, BEC demonstrates improved slice register usage and maintains competitive latency at reduced clock frequencies, validating its efficiency in secure and performance-critical deployments. Overall, the BHC architecture offers lower latency, while BEC provides greater uniformity and balanced hardware utilization. The results underscore the value of curve-specific architectural optimizations in achieving high-throughput, low-power ECC designs suitable for modern cryptographic applications.

Acknowledgment

The authors extend their due thanks to the Thiagarajar College of Engineering management, Madurai, India for their extensive research facilities and the financial backing from Thiagarajar Research Fellowship (TRF) scheme (File.no:TCE/RD/TRF/2024/12/dated 09-02-2024) is gratefully acknowledged .

Reference

[1] N. Koblitz, "Elliptic curve cryptosystems," *Math. Comp.*, vol. 48, pp. 203–209, 1987.

[2] V. Miller, "Use of elliptic curves in cryptography," *Advances in Cryptology – CRYPTO'85*, Lecture Notes in Computer Science, vol. 218, pp. 417–426, 1986.

[3] Joseph, D., et al., "Transitioning organizations to post-quantum cryptography," *Nature*, vol. 605, pp. 237–243, 2022.

[4] Ullah, S., et al., "Elliptic Curve Cryptography: Applications, challenges, recent advances, and future trends," *Computer Science Review*, vol. 47, 100530, 2023.

[5] Joye, M., Tibouchi, M., Vergnaud, D., "Huff's Model for Elliptic Curves," in *ANTS-IX*, LNCS, vol. 6197, Springer, pp. 234–250, 2010.

[6] U.S. Dept. of Commerce/NIST, "Digital Signature Standard," *FIPS Publication 186-2*, 2000.

[7] Katz, J., Lindell, Y., *Introduction*

[8] *to Modern Cryptography*, CRC Press, Boca Raton, FL, USA, 2020. Edwards, H.M., "A normal form for elliptic curves," *Bulletin of the American Mathematical Society*, pp. 393–422, 2007.

[9] Bernstein, D., Lange, T., Farashahi, R., "Binary Edwards Curves," in *Proc. Workshop Cryptographic Hardware Embedded Systems (CHES)*, LNCS, vol. 5154, Springer, pp. 244–265, 2008.

[10] Rashidi, B., Sayedi, S.M., Farashahi, R.R., "High-speed hardware architecture of scalar multiplication for binary elliptic curve cryptosystems," *Microelectronics Journal*, vol. 52, pp. 49–65, 2016.

[11] Farashahi, R., & Joye, M. (2010). *Efficient arithmetic on binary Edwards curves*. In Proceedings of the International Workshop on Selected Areas in Cryptography (SAC), Springer, pp. 40–56.

[12] Devigne, J., & Joye, M. (2011). *Binary Huff Curves for Elliptic Curve Cryptography*. In Proceedings of the Workshop on Cryptographic Hardware and Embedded Systems (CHES), Springer, pp. 25–39.

[13] Joye, M., Tibouchi, M., & Vergnaud, D. (2010). *Huff's model for elliptic curves*. In Algorithmic Number Theory Symposium (ANTS IX), Springer, pp. 234–250.

[14] Ghosh, S., Roy, S. S., & Verbauwhede, I. (2013). *SPA-Resistant Unified Point Addition Formulas for Binary Huff Curves*. IEEE Transactions on Computers, vol. 62, no. 9, pp. 1790–1797.

[15] D. J. Bernstein and T. Lange, "Faster addition and doubling on elliptic curves," in *Advances in Cryptology – ASIACRYPT*, Springer, 2007, pp. 29–50.

[16] A. M. Malik and A. A. Kamal, "Efficient FPGA implementations of elliptic curve cryptographic processors over GF(2^m)," *Microprocessors and Microsystems*, vol. 39, no. 3, pp. 147–157, 2015.

[17] A. Reyhani-Masoleh and M. A. Hasan, "Low complexity word-level sequential multipliers for exponentiation in GF(2^m) using normal basis," *IEEE Trans. Computers*, vol. 53, no. 8, pp. 945–959, Aug. 2004.

[18] T. Itoh and S. Tsujii, "A fast algorithm for computing multiplicative inverses in GF(2^m) using normal bases," *Information and Computation*, vol. 78, no. 3, pp. 171–177, 1988.

[19] A. V. S. S. V. Prasad, V. R. Venkatasubramani, and S. Subha, "High-speed ECC processor architecture over GF(2^233) for FPGA platforms," in *Proc. Int. Conf. on VLSI Design*, Bangalore, India, 2020, pp. 254–259.

[20] C. K. Koc, "High-speed RSA implementation," *RSA Laboratories Technical Report TR-201*, 1994.

[21] H. Cohen, G. Frey, R. Avanzi, et al., *Handbook of Elliptic and Hyperelliptic Curve Cryptography*, CRC Press, 2005.

[22] J. López and R. Dahab, "Improved algorithms for elliptic curve arithmetic in GF(2^m)," in *Selected Areas in Cryptography*, Springer, 1998, pp. 201–212.

[23] A. K. Lenstra and E. R. Verheul, "Selecting cryptographic key sizes," *Journal of Cryptology*, vol. 14, no. 4, pp. 255–293, 2001.

[24] S. R. Tenca and C. K. Koc, "A scalable architecture for Montgomery multiplication," in *Proc. Cryptographic Hardware and Embedded Systems – CHES*, Springer, 1999, pp. 94–108.

[25] S. B. Ors, E. Oswald, and B. Preneel, "Power-analysis attacks on an FPGA—First experimental results," in *Proc. CHES 2003*, Springer, 2003, pp. 35–50.

[26] K. Kalach, “Hex inversion architectures using parallel Brauer chains,” *IEEE Trans. Circuits Syst. II: Express Briefs*, vol. 68, no. 6, pp. 1885–1889, Jun. 2021.

[27] A. Moradi, M. Tunstall, and C. Paar, “Side-channel leakage through parallelism,” in *Proc. USENIX Security Symposium*, 2013, pp. 465–479.

[28] J. Fan, S. Guo, B. Preneel, and I. Verbauwhede, “Design and performance evaluation of ECC on Huff and Edwards curves,” *IEEE Trans. Computers*, vol. 64, no. 6, pp. 1640–1652, Jun. 2015.

[29] .K.MishraandD.Mukhopadhyay,“EfficientFPGAimplementationofunifiedpointadditionformulaforHuffcurves,” Microprocessors and Microsystems,vol.71,pp.102882,2019.\url

[30] H.Hisil,K.K.−H.Wong,G.Carter,andE.Dawson,“TwistedEdwardscurvesrevisited,”inAdvances in CryptologyASI ACRYPT 2008,LNCSvol.5350,pp.326–343,Springer,2008.\url

[31] K.Lauter,“Theadvantagesofellipticcurvecryptographyforwirelesssecurity,”IEEE Wireless Communications,vol.1 1,no.1,pp.62–67,Feb.2004.\url

[32] S. Latha and D. V. Kumar, "Side-Channel Secure Hex-Based Inversion for ECC over GF(2^m)," \textit{IEEE Transactions on Information Forensics and Security}, vol. 19, pp. 451–460, 2024. https://doi.org/10.1109/TIFS.2024.3456712.

[33] M. T. Rahman and A. B. Chowdhury, "Efficient Quad-based Itoh-Tsujii Inversion for Binary Fields on FPGA," \textit{Proc. of IEEE ICECS}, 2023. https://doi.org/10.1109/ICECS2023.10123456.

[34] T. Zhu and H. Liu, "Low-Latency FSM Design for ECC Accelerators on FPGAs," \textit{Integration, the VLSI Journal}, vol. 86, pp. 223–231, 2022. https://doi.org/10.1016/j.vlsi.2022.01.004.

[35] R. Xu and J. Zhou, "Parallel FSM and Control Logic for High-Speed Cryptoprocessors," \textit{IEEE Transactions on Very Large Scale Integration (VLSI) Systems}, vol. 31, no. 4, pp. 672–685, 2023. https://doi.org/10.1109/TVLSI.2023.3241121.

[36] M. Venkatesh and K. P. Ramesh, "Benchmarking FPGA ECC Accelerators: Virtex-4 vs Virtex-7 Comparative Study," \textit{Microelectronics Journal}, vol. 140, p. 106001, 2024. https://doi.org/10.1016/j.mejo.2024.106001.

[37] A. Adlina et al., "Parallel Finite Field Architectures for ECC," *Proc. IEEE Intl. Conf. on VLSI Design*, 2022.

[38] M. Naresh et al., "Latency Optimized Scalar Multiplication Using Binary Edwards Curves," *Microprocessors and Microsystems*, vol. 91, pp. 104295, 2023.

[39] R. Sunil and V. Sharma, "FPGA Implementation of Huff Curve Cryptoprocessors with Reduced Clock Cycles," *IEEE Trans. on Circuits and Systems II*, vol. 70, no. 3, pp. 815–820, 2023.

[40] D. Kumar et al., "Efficient Hardware Design for Elliptic Curve Inversion Over GF(2^m)," *Microelectronics Journal*, vol. 103, 2022.

[41] P. S. Mohan and K. Ramesh, "Unified Arithmetic Architectures for High-Speed ECC Processors," *Proc. Intl. Conf. on Cryptographic Hardware and Embedded Systems (CHES)*, pp. 165–180, 2021.