Comput Syst Sci & Eng (2017) 5: 337–351

International Journal of
Computer Systems
Science & Engineering

Design of Optimal Elliptic Curve Cryptography by using Partial Parallel Shifting Multiplier with Parallel Complementary

S Hemalatha[1],*; V Rajamani[2] and V Parthasarathy[3]

[1]*Assistant Professor,Department of Computer Science and Engineering, Vel Tech High Tech Dr Rangarajan Dr Sakunthala Engineering College, Avadi, Chennai 600062, Tamil Nadu, India. Tel.: +914426840249.*
[2]*Professor,Department of Electronics and Communication Engineering, Vel Tech Multi Tech Dr Rangarajan Dr Sakunthala Engineering College, Avadi, Chennai 600062, Tamil Nadu, India. Tel.: +914426840249.*
[3]*Professor,Department of Computer Science and Engineering, Vel Tech Multi Tech Dr Rangarajan Dr Sakunthala Engineering College, Avadi, Chennai*

High-speed Elliptic Curve Cryptography (ECC) modules implementation with less time, area, devices used is the recent research studies in cryptographic technology. The inclusion of modulus function in ECC contains several computations that lead to more time consumption and less speed. More multipliers and dividers utilization in ECC-based encryption/decryption required huge logic gates and registers. Hence, area and power consumption are more in ECC hardware unit. To overcome these problems, an enhanced ECC proposed in this paper. The framework optimizes the size of the multiplier and divider leads to a reduction of logic gates utilization. In our proposed work, we present a novel design structure for multiplier and divider based on ECC model defined by$x_n = mod\,(x * x, m)$. During the encryption, the self-multiplication of the input data for n number of times creates the public key. The remainder obtained from modulo division is regarded as the corresponding encrypted form of input data. To perform these multiplication and modulo division, we present a novel Partial Parallel Shifting Multiplier (PPSM) and Parallel Complementary Method (PCM) to reduce the number of logic gates and improve the operational speed. The comparative analysis between the proposed (PPSM-PCM) with the existing ECC architectures regarding the parameters such as number of logic gates, LUTs, FFs, frequency, power consumption, delay rate and latency assures the suitability of high-speed ECC in real-time applications.

Keywords: Elliptic Curve Cryptography (ECC), Look-Up Table (LUT), Parallel Shifting Multiplier, Parallel Complementary Method, Private Key, Public Key, Vedic Multiplier (VM)

1. INTRODUCTION

High-speed multipliers design to support the Digital Signal Processing (DSP) applications (transform, filtering), key generation in cryptography techniques is an active research area for the last decades. Real-time multiplier architectures focus the reduction in time, area and power consumption since the optimum architectures are preferred for multiplication. Vedic mathematics-based multiplier architecture is one of the optimized architectures that consume the less computation time and power. But, a number of logic devices utilization for simultaneous addition and partial product increases the occupation spaces. Hence, the replacement of adders with the Vedic-based multipliers and compressors is performed to multiply the 3-bits at a time. Thereby, the numbers of gates are reduced that reduces the occupational space. Vedic multiplier extends the traditional multipliers into reconfigurable designs that include the combination of Floating Point Multipliers (FPM) and unsigned multipliers. This paper extends the FPM applicability to cryptographic techniques to provide the secure Wireless Sensor Network (WSN) environment.

The evolution of paperless office strategies processes the documents in electronic platforms. A secure infrastructure is the pre-requisite for electronic communication systems. The mathematical tool to secure the data from the unauthorized access

*Corresponding author.

is referred as cryptographic techniques. Secure data transmission and the unauthorized access prevention are the important objectives of cryptographic techniques. The major categories of cryptographic techniques are public and private key-based encryption/decryption algorithms. For real-time applications, the public key establishes the private key for secure channel and the symmetric system provides the secure communication with the high-throughput.

The sub-exponential algorithm utilization in hybrid key-based cryptography introduces the Discrete Logic Problem (DLP) that initiates the Elliptic Curve Cryptography (ECC). The computational operation of ECC includes the single and double scalar multiplications leads to more non-zero digits. The generated key is expired after some time, a new key is generated by using the public key algorithm. The slowness of the public key cryptographic protocols requires the hardware support. When ECC is implemented with the Field Programmable Gate Array (FPGA) - based hardware, the point multiplication is crucial due to LUT complexity and time-area constraints.

Hence, the research works focus the mathematical optimization of cryptographic primitives on the basis of ECC. The mathematical formulation of point multiplication with field basis (Binary Edwards Curve (BEC) and Generalized Hessian Curves (GHC)) is a delay-oriented process. Traditional works addressed the following limitations in ECC processors design: more area and power consumption due to large multiplication. When ECC is implemented with the Field Programmable Gate Array (FPGA), the point multiplication is crucial due to LUT complexity and time-area constraints. To overcome these problems, an enhanced ECC with Partial Parallel Shifting Multiplier (PPSM) and Parallel Complementary Method (PCM) are implemented in this work. The technical contributions of proposed PPSM-PCM are listed as follows;

- The parallel shifting-based multiplier construction offers less number of computations that leads to better resource utilization with minimum clock cycles.
- The division of large multiplication unit into smaller blocks via Parallel Complementary Method (PCM) improves the operating frequency.
- The combination of Partial Parallel Shifting Multiplier (PPSM) and Parallel Complementary Method (PCM) in this paper effectively reduces the number of logic gates and storage space compared to the existing methods.

The main innovations of proposed PPSM-PCM are highlighted as follows:

- FPGA-based ECC processors discussed in this paper utilizes the PPSM-PCM-based multiplication and division operation improves the encryption / decryption performance.
- Reduction of multiplications by shifting operation reduces the gates utilization that enhances the system availability and reduces the overall cost
- The minimization in data path causes the reduction in delay and power consumption compared to reduced significant digits-based ECC processors.

The paper organized as follows: The detailed description of the related works on the multiplier architecture and its FPGA implementation in section 2. The implementation optimal multiplier, algorithm required for construction and clock cycle analysis in section 3. The performance analysis on device parameters presented in section 4. Finally, the conclusions about the application of proposed ECC presented in section 5.

2. RELATED WORK

This section presents the detailed description of the evolution of fast FPGA-multiplier architectures for the Elliptic Curve Cryptography (ECC) to minimize the area, time and occupational space.

2.1 Cryptographic techniques

The proper trade-off between the parameters namely, speed, area, time is an important requirement for real-time architecture design. *Aneesh and Mohan* (1) analyzed various adders to select the best adder to reduce the critical path delay and provide a balance between area, time and speed constraints. The Fast Fourier Transform (FFT) is highly preferable and its reconfigurable hardware capability raised the cryptographic mechanisms. *Poppelmann and Guneysu* (2) presented the lattice-based cryptography for an efficient FFT computation under reconfigurable. The time consumption of extension fields was more. *Morales et al* (3) explored the Linear Feedback Shift Registers (LFSR) to reduce the time consumption. Encryption algorithms govern the secure data exchange which reduced the area and power consumption. A secure, high-quality algorithm called RSA which is used for health record preservation (4). *Sahu et al* (5) presented the architecture and modeling of modular multiplication on the basis of RSA algorithm. The time-consuming process in RSA encryption / decryption is an estimation of “ab mod n”. The tedious computation made the hardware as complex. *Bhaskar et al.* (6) reduced the hardware complexity by the elimination of shifting operation. The sub-exponential algorithm utilization in the RSA encryption model introduced the problem called Discrete Logic Problem (DLP).

2.2 Influence of Elliptic Curve Cryptography

To overcome DLP and reduce the modular function complexities, research works introduced the Elliptic Curve Cryptography (ECC) which includes single and double end scalar multiplications. *Adikari et al.* (7) used the integer coding techniques to reduce the density of non-zero digits. *Azarderakhsh et al.* (8) presented the Binary Edwards Curve (BEC) and Generalized Hessian Curve (GHC) based point multiplication under Gaussian basis. *Rebeiro et al.* (9) reduced the clock cycles and increased the operating frequency to speed up the scalar multiplications. They presented the scheduling schemes for clock cycle reduction. *Sasdrich et al.* (10) realized the ECC instance curve 25519 by a reconfigurable hardware by DSP-based single core and extended architectures with dedicated inverter stage lead to a number of multiplications. *Tseng et al.* (11) investigated the hierarchical and dynamic ECC influence via message

overhead analysis. The fastest multiplier is the pre-requisite for reconfigurable hardware. *Roy et al.* (12) illustrated the suitable scheduling for performing point multiplication and doubled the pipeline stages to design the fastest multiplier. The architectural and timing constraints required the recorder and reorganization in GF filed.

2.3 Vedic mathematics-based multiplier

The fundamental operation of FPGA applications like transform, convolution, and filtering processes is multiplication. *Kumar and Charishma* (13) designed the high-speed multiplier by using ancient Indian Vedic Mathematics. They enabled the parallel intermediate product generation, which eliminates the reduction of unwanted multiplication steps. *Haveliya et al.* (14) discussed the block convolution process on the basis of the vertical and crosswise algorithm. The arrival of new technologies in VLSI platform demanded the high-speed low area occupation. *Rao et al.* (15) introduced the compression based multiplier architecture for high-speed applications on the basis of Vedic mathematics. *Subhuthi et al.* (16) dealt with the implementation of the Vedic divider to satisfy the reduction of time delay and area occupation and improved the operation speed. *Bathija et al.* (17) proposed the 16×16 multiplier with the Urdhva Tiryakbhyam Sutra in Vedic mathematics for cryptographic applications in WSN that initiates the extension field-based multiplier implementation.

2.4 Binary extension field-based multiplier design

Designing of multipliers on binary extension fields faced the number of difficulties such as different sized elements, more area occupation. *Uslu et al.* (18) prevented these difficulties by a combination of sparse irreducible polynomials and unification of modular reduction. There are different computational levels of multiplication in the data path specified by binary extension field and Galois Field (GF). *Sutter et al.* (19) found the optimal digit size by GF multiplication and division. They achieved the multiplication in required time. The most critical application in ECC is Elliptic Curve Point Multiplication (ECPM). *Esmaeildoust et al.* (20) presented the hardware architecture of ECPM based on Residual Number System (RNS). They presented the pipelined architecture to improve the speed of multiplication operation. *Mehdizadeh et al.* (21) reorganized the critical path for Lopez-Dahab into the non-critical path, which reduced the resource consumption with extreme constraints. *Azarderakhsh et al.* (22) utilized the Gaussian Normal Bias (GNB) over GF binary extension field. They introduced new architecture of affine coordinate based point addition.

2.5 Integration of FPGA, ECC, Extension fields

Marzouqi et al. (23) reviewed the design options, finite extension fields and the target systems required for hardware implementation of ECC. The difficulty of ciphertext creation in both encryption and decryption algorithms is more. *Kamalakannan et al.* (24) discussed the FPGA-based implementation of ECC multiplier in which the transformation of text message into the affine point on ECC over GF. *Marin et al.* (25) focused the mathematical formulation of key primitives, authentication and integrity computations. The regular structure of Vedic Multiplier (VM) has more delay and area leads to the Floating Point Multiplier (FPM) design. *Anjana et al.* (26) proposed the design of high-speed FPM. They utilized different adders like ripple carry and carry look ahead for the design of single precision FPM. *Chatterjee et al.* (27) extended the curve of ECC to Binary Edward Curve (BEC) for GF(2^{233}). They performed the simple power analysis by the naïve implementation.

2.6 ECC hardware

The hardware implementation of Floating Point Multiplier (FPM) based on ECC cryptography was difficult. *Rajagopalan et al.* (28) presented the survey of hardware implementation algorithms and techniques for cryptographic implementation. *Kumar et al.* (29) utilized the complex instructions to reduce the latency of ECC implementation. The point doubling and addition were performed by using the Lopaz-Dahab algorithm and the inversion at the last stage were performed by using the Itoh-Tsujii algorithm. *Li* and*Liu* (30) discussed the application of ECC to the security improvement in WSN. They discussed the authentication and key-based protocols-based security development in WSN. *Abdulrahman* (31) discussed the parallelism algorithm that extended the twisted Edwards curve to achieve the scalar multiplication. *Azarderakhsh et al.* (32) proposed an efficient and high-performance curve based (BEC and GHC) point multiplication. They modified data flow analysis to predict the latency. The combination of dual Processing Element (PE) with the priority scheduling achieved the power-analysis-resistant field. (33). *Liu et al.* (34) presented the Montgomery ladder method based on the polynomial Finite Field (FF) arithmetic to improve the performance and speed on the FPGA-based ECC hardware. They compared its performance with the traditional architectures in (35), (36), (37), (38), (39) and (40). *Sghaier et al.* (41) investigated the Area-Time efficient hardware model for ECC system by using EC-Discrete Logarithmic Problem (EC-DLP). The trade-off between the less device utilization and high-data security is the important requirement for ECC hardware.

3. OPTIMAL ECC ARCHITECTURE

The assurance of data confidentiality is provided by mapping process of a text message to points on the elliptic curve. The ECC based encryption involves a number of scalar or double multiplications and modulo divisions consume more area, power thereby the delay is increased. Hence, the reduction of computations in multiplication leads to less resource, area utilization offers maximum operating frequency and speed. In this paper, an optimized ECC is proposed for computational reduction. Fig. 1 shows the overall architecture of Elliptic Curve Cryptography (ECC).

The architecture of 163-bit ECC includes two major parts as follows:

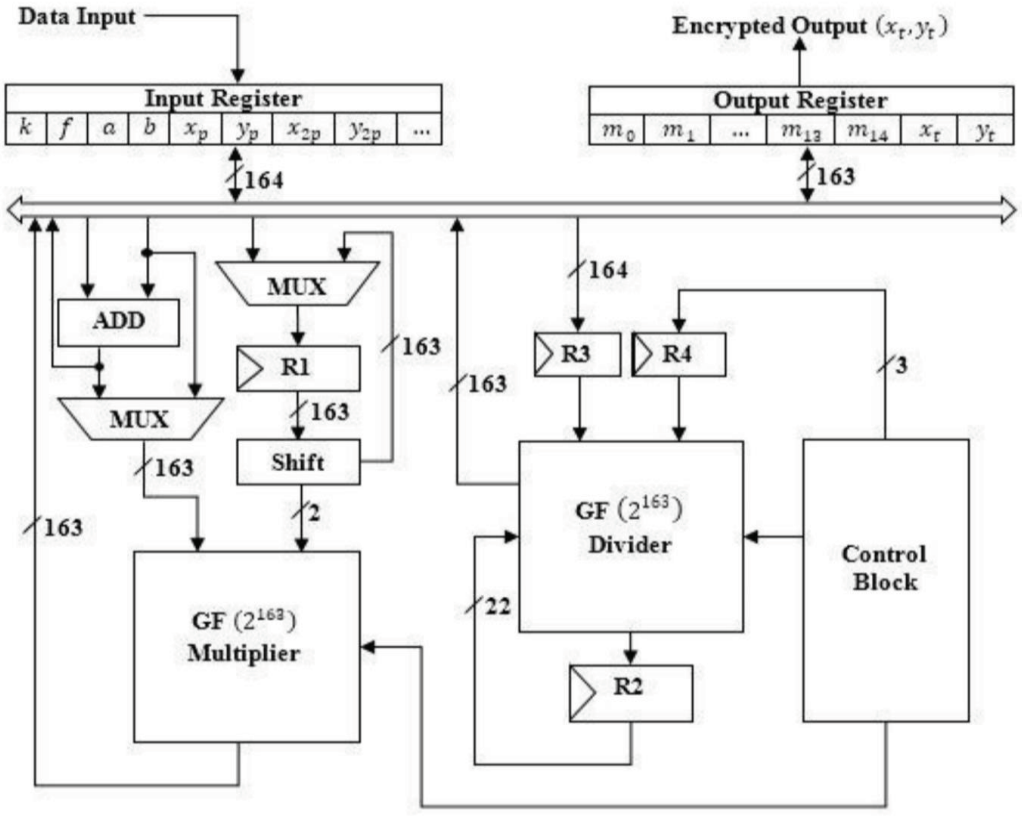

Figure 1 Architecture of 163-bit ECC.

- Parametric registers
- Temporary registers

The 163-bit ECC cryptographic engine includes 3X finite field multiplication, finite field division. The Galois Field (GF)-based multiplier design includes the polynomials as follows:

$$A(x) = \sum_{i=0}^{m-1} a_i x^i \tag{1}$$

$$B(x) = \sum_{i=0}^{m-1} b_i x^i \tag{2}$$

Where, m-dimension and i- each value

The multiplied output is represented as

$$z(x) = A(x) \sum_{i=0}^{m-1} b_i x^i \tag{3}$$

The realization of 3X faster GF multiplier is mathematically modeled as

$$z_{3k} = \left[b_0 A(x) + b_3 x^3 A(x) + \ldots \right] \mod P(x) \tag{4}$$

Where, $P(x)$ denotes the elliptic curve over the point x

The finite division formulation depends on the $A(x)$ times the inverse values of $B(x)$. The table maintenance and the algorithm-based methods govern the division process. For the mall values of m, the table maintenance approaches are efficient. But, the increase in data dimension increases the circuit size. The reduction in number of iterations is the major focus of algorithm-based methods. The division unit contains more number of cells. The finite field division operation includes the following operations

- Assign the input dividend and divisor into the cells
- Each cell in the ECC architecture performs the division operation depends on the output signals.
- Upon the cycle completion, the divider yields the output in each cell.

The parameters for the ECC implementation in this work are base point and the irreducible polynomial represented as

$$p(x) = x^{163} + x^{15} + 1 \tag{5}$$

The ECC parameters such as key k coefficients (a, b, f), the point to be encrypted are (x_p, y_p) and the scalar multiplication values are preloaded to the parameter registers to reduce the time consumption. The results from the scalar multiplication are stored in the intermediate registers. The radix-4 redundant recoding applied on the binary representation of ECC reduces the number of computations that reduces the clock cycles effectively.

3.1 Optimal ECC

The architecture of proposed optimal multiplier is shown in Fig. 2. At first, the inputs (multiplier and multiplicand) are received

as a binary stream and stored in the registers R1 and R2 respectively. The Elliptic Curve Cryptography (ECC) based multiplication model described as follows:

$$x_n = mod\,(x * x, m) \tag{6}$$

Here, the binary encoded data denoted by 'x' multiplied together and perform modulo division for 'n' number of iteration denoted by public/private key. The output of multiplier architecture is checked whether it is greater than maximum size 'm'. The public and private key for the proposed model are chosen as 29 and 5 respectively. During encryption process, the input binary coded data is multiplied itself and modulo division is performed. The remainder is stored in the third register 'R3' up to the value of public key. The encrypted output form for the input is retrieved from the register 3.

The proposed architecture of ECC based encryption of input data consists of following steps.

- The maximum size of encrypted data is fixed initially.
- The input binary data is multiply by itself repeatedly.
- Check whether the output of the multiplier is lesser than the maximum size.
- If it is greater than the maximum allowable size, then modulo division is performed and store the remainder in a separate register.
- If the multiplier output is greater than the maximum size, then the division process is repeated.
- The remainders in the registers are the required encrypted form of the input data.

3.2 Multiplier

In general, the $N \times N$ multiplication process generates the number of partial products which are added up to produce the final output. The increase in the size of multiplier increases the number of partial products (e.g. 4×4 multiplication requires 16-bit partial products, 8×8 requires 64-bit partial products, etc.). Briefly, the f-number of increase in bits increases the partial products as f^2. Fig. 3 shows the multiplication of two 8 bit numbers. The number of shifting and addition are more which consume more time. The shifting process requires the latency of 8 clock cycles reduces the operating frequency.

But, the performance of arithmetic units required the optimal resource utilization. To achieve this, large multiplication can be divided into smaller ones and then added up which makes better time consumption and resource utilization as shown in Fig. 4. The operation of multiplication by smaller units in Fig. 4 performed with the minimum latency of 3 clock cycles compared to existing one which increases the operating frequency.

3.3 Modulo Divider

Modulo division performed in this paper is based on parallel complementary method. The flow of modulo division is shown in Fig. 5.

Initially, the dividend is stored in register R2, the 2's complement of the divisor is taken and stored in register R1. The contents of registers of $R1$ and $R2$ are added together by an adder. The sum is stored in register $R3$, The carry is loaded in register $R4$. Then, check the value of register $R3$ is less than or equal to the divisor. The remainder and the quotient are stored in register $R5$ and $R6$ if the condition is satisfied.

3.4 PPSM-PCM based ECC algorithm

There are three algorithms such as ECC, multiplier, and division proposed in this paper described in this section. The input message data, key pairs (public and private key) are given as the inputs to the ECC algorithm. The algorithm for ECC as follows:

Proposed ECC Algorithm
Input:Input message data, 'D', Public Key, 'Pb', Private Key, 'Pr'
Output:Encrypted message data, 'E', and Decrypted Message date, 'Dm'
Step 1: Load input data to Register, R1 = D;
Step 2: Load internal Register, R2 = D;
*Step 3: Multiply two Register values, R3 = R1 * R2;*
Step 4: If (M > G) //Check whether the Multiplied value greater than maximum limit or not
Step 5: MD = mod (M, G); //Modulo Division of 'M' and 'G'
Step 6: R4 = MD;
Step 6: Else
Step 7: R4 = M;
Step 8: End If
Step 9: Update R2 = R4;
Step 10: Repeat Step 3 to 9 until iteration time of 'Pb'
Step 11: R5 = R4; E = R5; //Encrypted Data
Step 12: R1 = E; R2 = E; //Update R1 and R2 Register
Step 13: Repeat Step 3 to 9 until iteration time of 'Pr'
Step 14: R5 = R4; Dm = R5; //Decrypted Data

Initially, the input data is loaded into the registers $R1$and $R2$.The multiplication performed on the contents of registers and stores the result in register $R3$. Then, Check whether the multiplied result is greater than the maximum value. Modulo division is performed for the value is greater than the size and stored the output *(MD)* in register $R4$. Otherwise, the output is loaded into the register $R4$. The new value in register $R4$ replaced the existing value in register R4. This processes repeated for the value of the public key is reached. Finally, the $R1$and $R2$ is updated with the encrypted data. The ECC process in this paper holds the optimal multiplier algorithm described as follows:

The inputs (*In1* and *In 2*) are given to the algorithm. The input 1 is stored in register $R1$, and the input 2 is stored in register $R2$. Then, split the contents of $R2$ in LSB and MSB bits and loaded into registers $R3$ and $R4$. After loading, the multiplication of contents is $R1$ with $R3$, $R1$ with $R4$. The partial products after shifting process ($R5$ and $R6$) are added together and stored in the output register ($R7$).

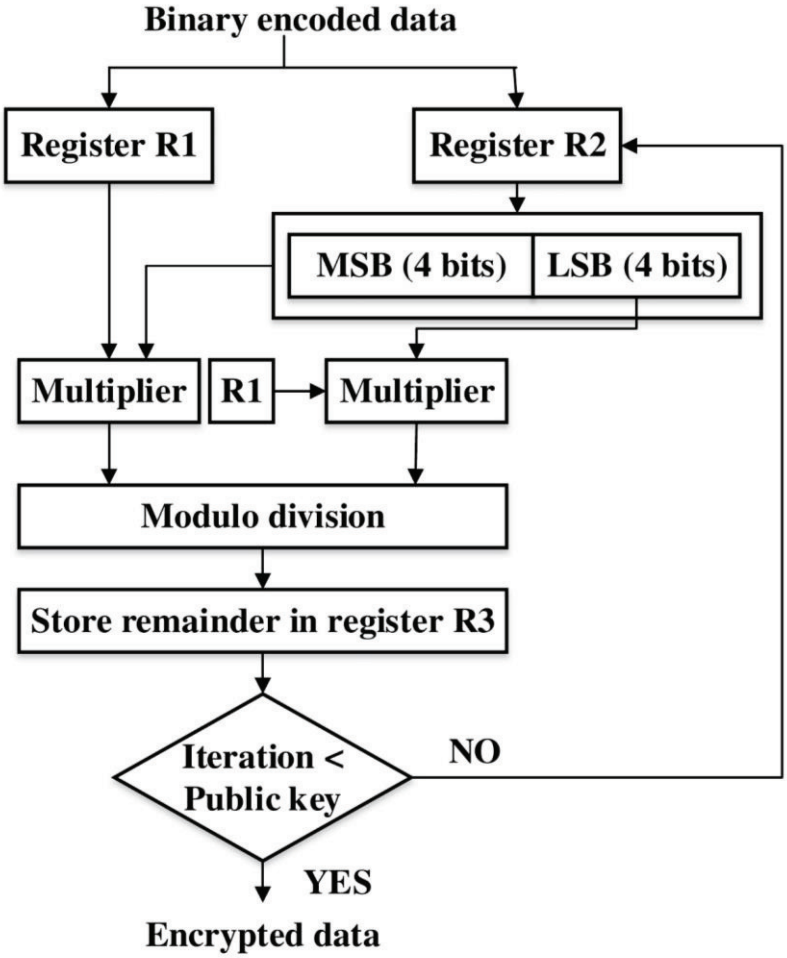

Figure 2 Flow diagram of optimized ECC.

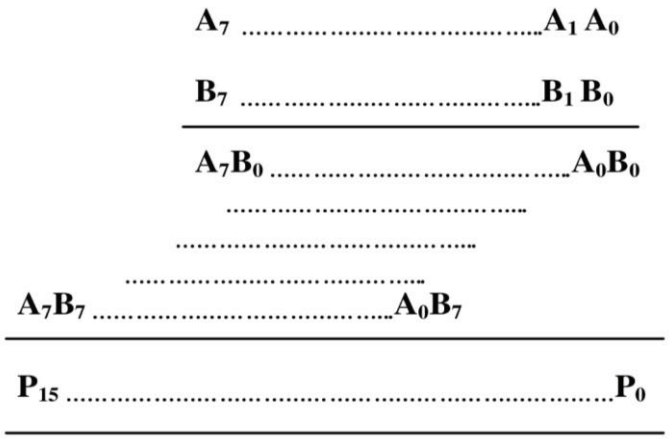

Figure 3 8×8 bit multiplication.

The complementary and parallel operation (PCM) of division operation performed in this paper offers the optimal reduction of computational steps which reduces the delay resource consumption. The algorithm of proposed modulo division is as follows:

Initially, loading of input values ($D1$, $D2$) to the registers ($R2$, $R1$) performed. The divisor is subject to 2's complement method and stored in the respective register. Then, check whether the remainder in $R3$ is less than or equal to divisor $D2$. The repetitive addition and updating process of remainder provides the required output.

3.5 Clock cycle analysis

This section presents the clock cycles required for the design of ECC model. During the positive edge of the clock pulse, the

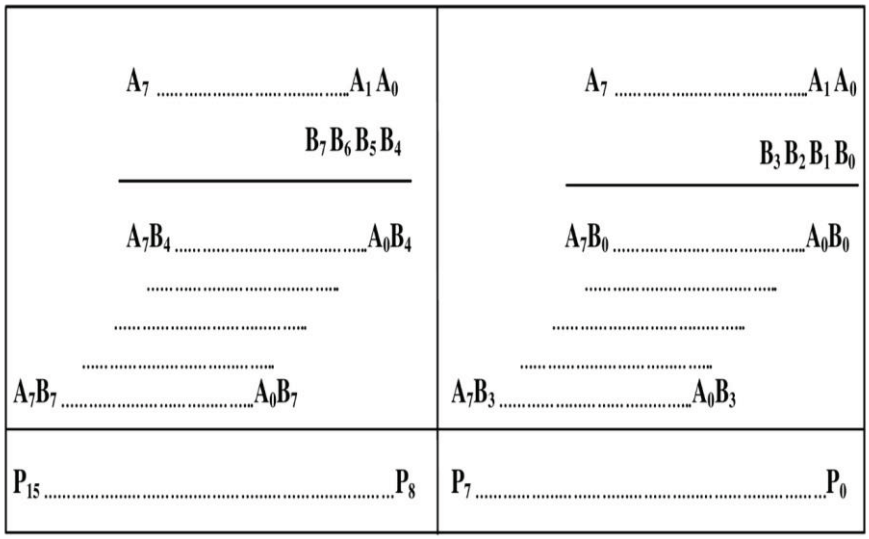

Figure 4 Large multiplication by smaller units.

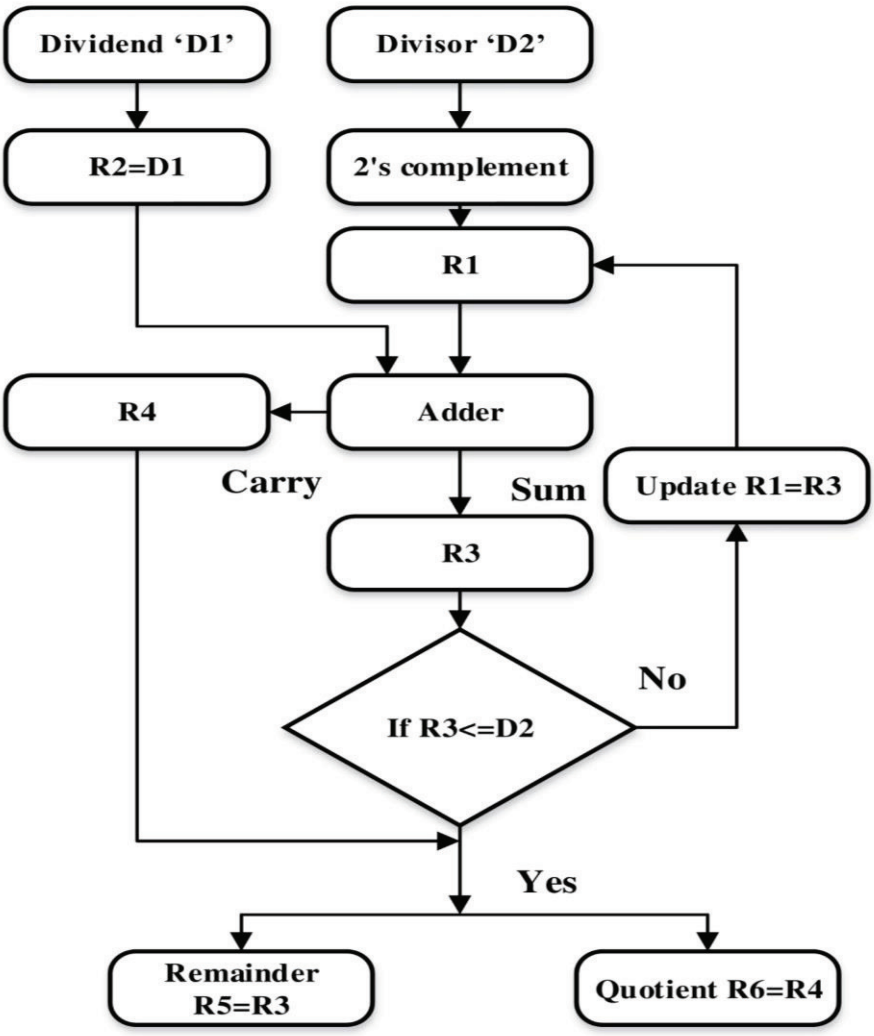

Figure 5 Flow of modulo division.

inputs1 and 2 changes the states from the previous one. The multiplication process in the initial stage of ECC requires three clock cycles for extracting the input 1 from R1 register, input 2 from the R2 register and multiplication. During multiplication process, the state of modulo division is in don't care condition denoted by the dotted lines. After the multiplication process,

PPSM based Multiplier Algorithm

Input:Input 8-bit data, 'In1' and 'In2'.
Output:Multiplied Result, 'P'
Step 1: R1 = In1; R2 = In2; //Load Input data to registers
Step 2: R3 = R2 [3:0]; R4 = R2 [7:4]; //Split data in two internal registers.
Step 3: R5 = R1 & R3; //AND operation between two bit streams
Step 4: R6 = R1 & R4; //AND operation between two bit streams
Step 5: Shift and add R5 and R6 to 1 bit size
Step 6: R7 = R5 [11:4] + R6 [7:0]; //Partial Adder.
Step 7: P [3:0] = R5 [0:3]; P [11:4] = R7; P [15:12] = R6 [15:12] //Output Register

PCM based Modulo division algorithm

Input: Input 8-bit data, Dividend, 'D1' and Divisor, 'D2'.
Output: Divided Result, Quotient, 'Q' and Reminder, 'Rm'.
Step 1: R2 = D1; //Load Input data to registers
Step 2: R1 = not (D2) + 1'b1 //Apply 2's Complement of Divisor, 'D2'
Step 3: R3 = Sum (R1, R2); //Add two registers, R1 and R2.
Step 4: R4 = Carry (R1, R2); //Extract Carry from adder Result.
Step 5: If (R3 <= D2)
Step 6: R5 = R3; //Reminder.
Step 7: R6 = R4; //Quotient.
Step 8: Else
Step 9: R1 = R3; //Update R1
Step 10: Repeat Step 3 to 5
Step 11: End If

modulo division is performed which consumes another three cycles (R4, R5, and R6). Finally, the output is stored in the respective register in R7 cycles. The analysis of clock cycle is described in Fig. 6.

The theoretical model is implemented in FPGA device Xilinx Virtex-4 XC4VLX160 FPGA. The simulated output is represented in Fig. 7. During 0 – 40ns, the input 1 and 2 are loaded into the register 1 and 2. The input data is multiplied by itself up to the value of the public key that consumes first three clock cycles. During multiplication, the outputs of division (Quotient, reminder) are don't care conditions (X, 0) respectively. After multiplication, modulo division is performed and the remainder is extracted for the next three cycles. The outputs are represented as don't care during multiplication and division processes. The cycle after modulo division, the storage of output is initiated.

The simulated output shows that the multiplication output is loaded to the register within the 8 clock cycle periods. The reduction in clock cycle improves the operating frequency. The FPGA implementation of ECC reduces the number of computational steps that reduces the storage overhead that leads to less power consumption.

4. PERFORMANCE ANALYSIS

This section presents the analysis of area utilization, delay and the operating frequency of proposed PPSM-PCM based ECC architecture on FPGA device. The on-chip power consumption for various blocks and the variation of power due to the junction temperature is discussed.

4.1 On-chip power

The on-chip power variation against junction temperature, processes and the internal operating voltage for optimized MQ coder architecture are discussed in this section. The consumption of power for each block in Virtex-4 XC4VLX160 architecture is depicted in Fig. 8.

From the Fig. 8, it is observed that the optimized probability of state estimation value reduces the number of bits by compression of redundant pairs reduced the memory consumption (Block RAM) and the number of Mixed Mode Clocked Managers (MMCM) required. The power consumption against voltage variation for fixed temperature (27°C) and power ratings against junction temperature for fixed voltage (1 V) respectively shown in Fig. 9 and Fig. 10.

The reduction of a number of bits from optimized coder gradually increases the total power consumption against internal operating voltage (Vcc) and junction temperature. The increase in junction temperature will increase the power flow (Static and total) in a chip. The power variation for specific temperature (50, 85 and 100) for typical (1.0 V) and maximum (1.03) voltage ratings illustrated in Fig. 11

4.2 Utilization analysis

The comparative analysis between the proposed Partial Parallel Shifting Multiplier (PPSM) with the existing ECC based multipliers such as Binary Edward Curve (BEC) (27) and Generalized Hessian Curve (GHC) (8) over the field of $GF(2^{163})$ and $GF\left(2^{233}\right)$ on the parameters of latency, maximum frequency, Area, time required for point multiplication and area time product. The device utilization for proposed PPSM-PCM on Xilinx Virtex-4 XC4VLX160 FPGA device for overall ECC architecture, 8 bit multiplier architecture, 8 bit modulo divider architecture is shown in Table 1, 2 and 3 respectively.

Table 1 Device Utilization Analysis (ECC Architecture).

Device Utilization Summary			
Logic Utilization	**Used**	**Available**	**Utilization**
No. of Slice Registers	141	93120	0%
No. of Slice LUTs	671	46560	1%
No. of fully used LUT-FF pairs	107	705	15%
No. of bonded IOBs	17	240	7%
No. of BUFG/BUFGCTRLs	1	32	3%

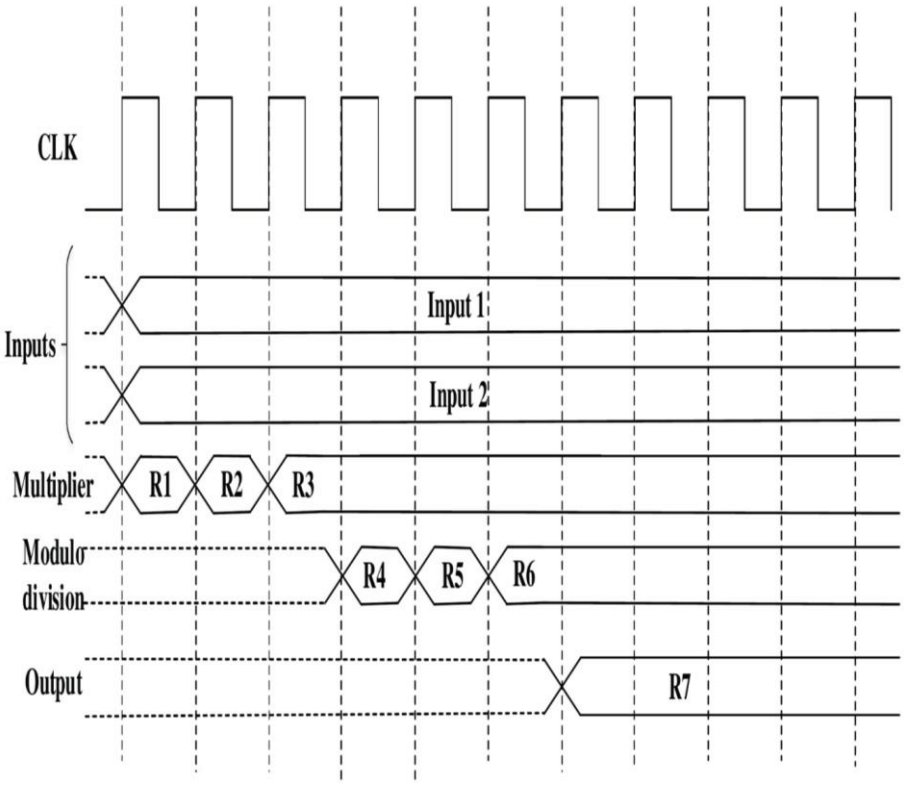

Figure 6 clock cycle analysis.

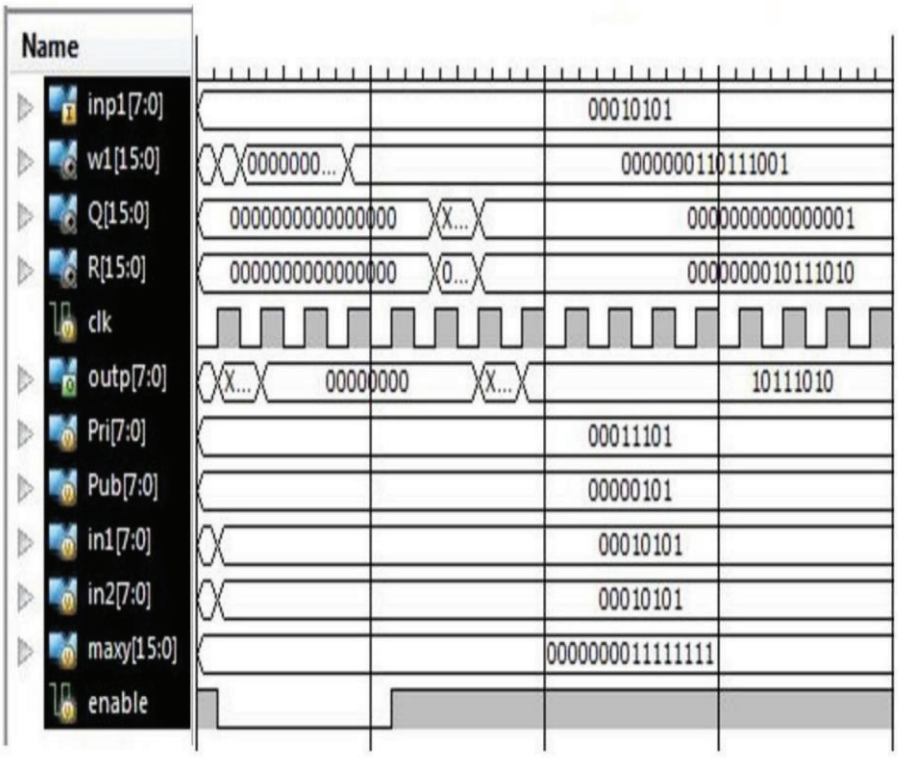

Figure 7 Simulation output.

The proposed PPSM is tested with Virtex 5 architecture and compared the result with existing architecture (18) in Table 4. The proposed PPSM offers significant reduction of 60.88 %, 78.53 % and 56.04 in FF, LUTs, and period compared to MSD multiplier. The optimized multiplier architecture used in ECC is much suitable for an encryption operation. The graphical representation of this comparison is shown in Fig. 12.

The comparative analysis of proposed PPSM architecture with the GFAU/MAS (32) for the field size of 163 and 233 listed in Table 5 and VI respectively. From the tables, the MAS provides

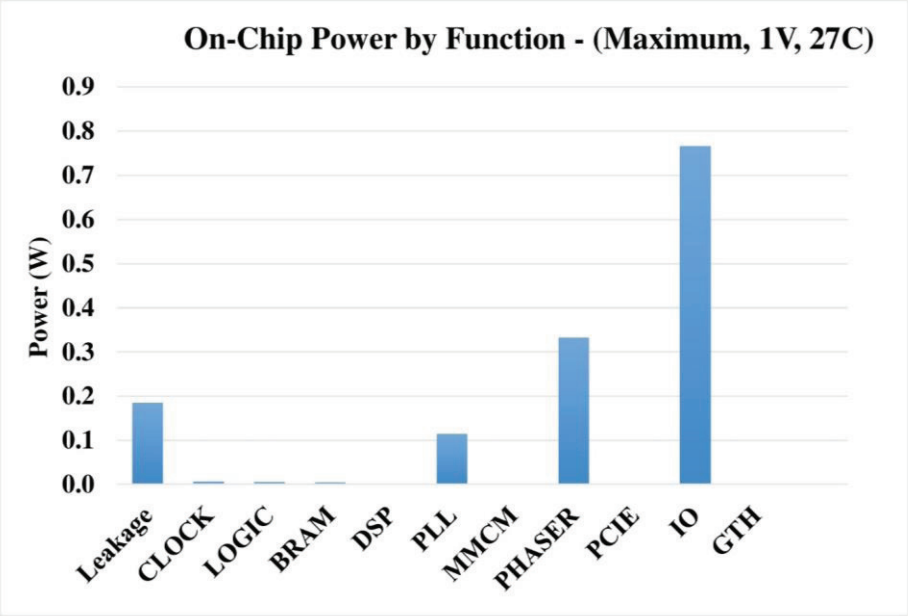

Figure 8 On-chip power vs. Blocks.

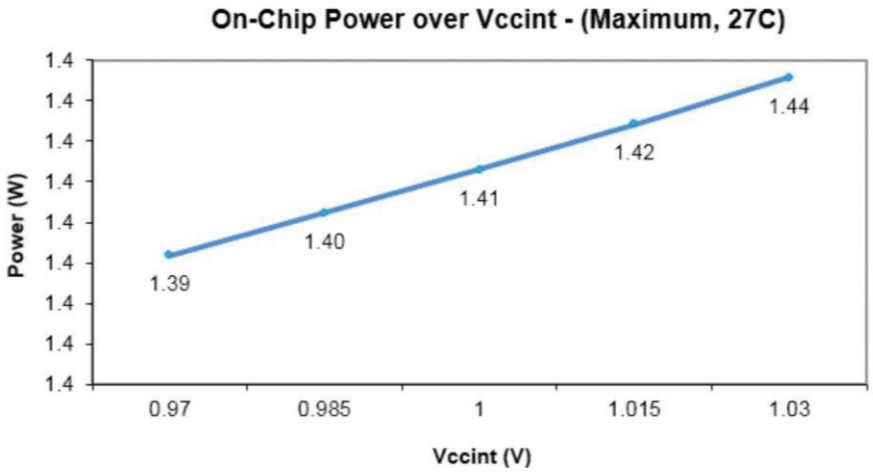

Figure 9 On chip power vs. Internal Vcc voltage.

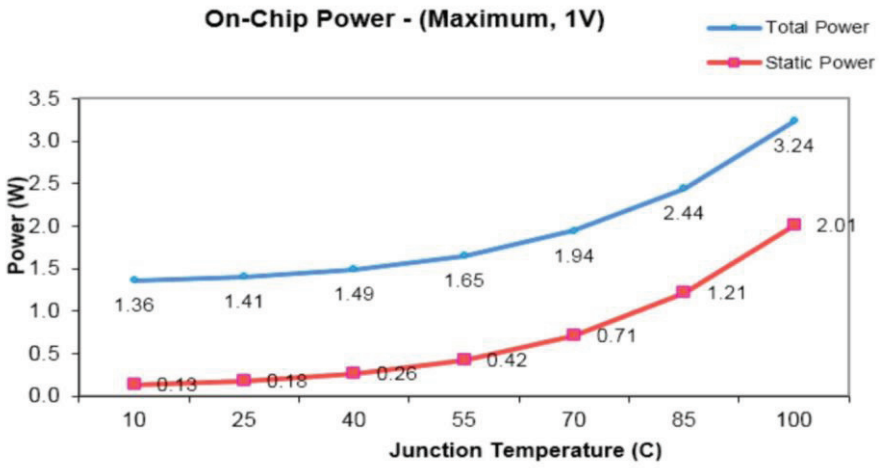

Figure 10 On-chip power vs. Junction temperature.

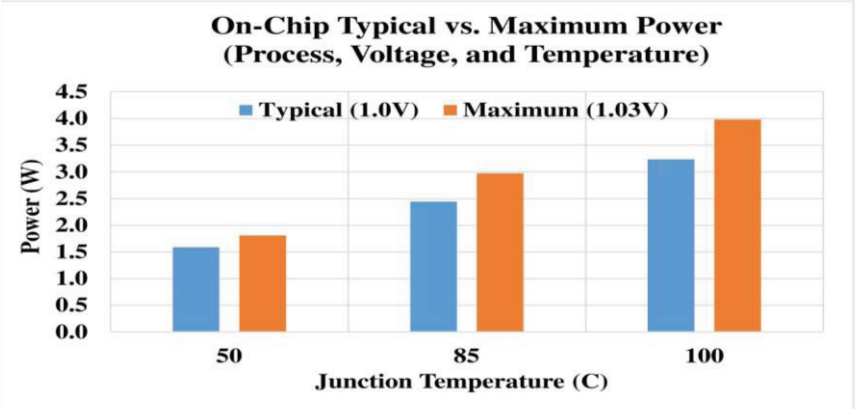

Figure 11 Typical, maximum power vs. Junction temperature.

Table 2 Device Utilization Analysis (8-Bit Multiplier).

Device Utilization Summary			
Logic Utilization	**Used**	**Available**	**Utilization**
No. of Slices	97	67584	0%
No. of Slice Flip Flops	72	135168	0%
No. of 4 input LUTs	171	135168	0%
No. of bonded IOBs	33	768	4%
No. of GCLKs	1	32	3%

Table 3 Device Utilization Analysis (8-Bit Modulo Divider).

Device Utilization Summary			
Logic Utilization	**Used**	**Available**	**Utilization**
No. of Slices	46	93120	0%
No. of Slice Flip Flops	575	46560	1%
No. of 4 input LUTs	34	587	5%
No. of bonded IOBs	66	240	27%
No. of GCLKs	2	32	6%

Table 4 Comparative Analysis.

Multiplier with $w = 8$ digit size	FF	LUT	Period (μs)
MSD 1st Multiplier (MSD)	1,728	3,186	3,224
Right Justified Operands, Separate Reduction (RJOSR)	1,744	3,239	3,987
Left Justified Operands, Separate Reduction (LJOSR)	1,756	3,248	3,688
Right Justified Operands, Unified Reduction (RJOUR)	1,744	3,487	3,896
Left Justified Operands, Unified Reduction (LJOUR)	1,756	3,477	3,891
PPSM	**676**	**684**	**1,417**

the minimal performance in latency (39.45 %) and the area-time product (13.27 %) for the field size 163 and for 233 size it provides 30 % and 52.48 % respectively.

The MAS also reduces the number of slices, LUTs, and FFs by 34.28, 4.8 and 44.44 % for the field size 163 and they are 29.92, 8.07 and 50.42 % for the field size 233. The parallel computation and the shifting operation in proposed PPSM-PCM further reduce the latency, area-time product, point multiplication time, slices, LUTs and FFs by 36.51, 91.33, 15.71, 1.13, 43.49 and 74 % respectively for field size 163. For the field size 233, the PPSM-PCM offers the reduction in terms of 30.57, 76.81, 15.74, 13.98 and 63.58 % in latency, point multiplication time, slices, LUTS, and FFs respectively.

4.3 Message overhead

The network performance is validated with the help of investigation of the number of messages transmitted for various sessions and Secure Access Points (SAP) in the Network Simulator (NS-2) environment. The message overhead to establish the secure environment is more in the traditional Hierarchical and Dynamic Elliptic Curve Cryptography (HiDE) (11) and RSA 1024 (4) key implementation.

Fig. 13 and 14 show the message overhead variations for different sessions and SAPs. The increase in sessions linearly increases the message overhead. For the minimum sessions (2), the message overhead for HiDE (0.8 kB) is less compared to the RSA 1024 (2 kB). But, the ECC implementation under PPSM-PCM framework reduces the overhead further (0.5 kB). Similarly, the PPSM-PCM overhead is 34.38 % less compared to the HiDE for the maximum sessions (10).

The increase in SAPs gradually decreases the message overhead. Compared to RSA 1024, the HiDE overhead values are 19 and 4 kB for minimum and maximum SAPs. The optimization provided in PPSM-PCM further reduces the overhead by 42.1 and 50 % compared to HiDE. The novel architecture for multiplier and modulo-divider architecture reduces the logic gates and the number of computations that provides the optimal trade-off between the area, time and power consumption. The reduction in a number of logic gates reduces the clock delay requirement that provides the high operating frequency.

Table 5 Comparative Analysis (Field Size = 163).

Implementation Result for Field Size, m = 163, d = 33, and q + 3 = 8							
Methods	**Latency**	**Fmax (Hz)**	**Area**			**Point M (μS)**	**AT (Area × Time)**
			Slices	**LUTs**	**FFs**		
GFAU	2,968	62	4,422	25,422	15,094	3.5	0.113
MAS	1,797	62	2,906	24,196	8,385	3.5	0.098
PPSM-PCM	1,141	247.1	2,873	13,672	2,180	2.95	0.0085

Table 6 Comparative Analysis (Field Size = 233).

Implementation Result for Field Size, m = 233, d = 26, and q + 3 = 12							
Methods	**Latency**	**Fmax (Hz)**	**Area**			**Point M (μS)**	**AT (Area × Time)**
			Slices	**LUTs**	**FFs**		
GFAU	3,356	57	5,528	24,713	17,256	4.2	0.157
MAS	2,349	58	4,874	22,719	8,556	4.2	0.0746
PPSM-PCM	1,631	222.39	4,107	19,543	3,116	4.22	0.0173

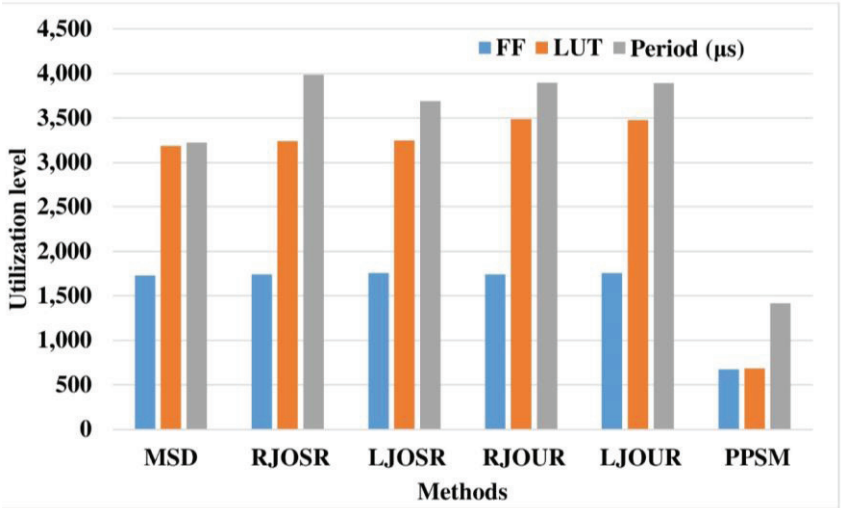

Figure 12 Utilization analysis for existing and proposed a method.

5. IMPLEMENTATION ANALYSIS

The effectiveness of the proposed work is investigated by extending the implementation analysis to the virtex-4 architecture under the various existing models. Compared to the traditional models, the Montgomery ladder (34) and ECDLP (41) provide the minimal area (10417, 9300) and time consumption (9 and 5.75) respectively for the field of 163.

The novel architecture for multiplier and modulo-divider architecture reduces the logic gates and the number of computations that leads to further reduction in slices and time consumption i.e. 2873 and 2.95 μs. The comparative analysis between the proposed PPSM-PCM with the existing ECDLP states that the proposed PPSM-PCM provides the 69.1 and 48.7 % reduction in slices and time consumption respectively.

6. CONCLUSION

This paper addressed the problems of the area, time and power consumption due to the utilization of a number of logic gates and registers for multiplication and division operations. To overcome these problems, the optimum hardware architecture of Elliptic Curve Cryptography (ECC) proposed in this paper. The optimization in a number of gates, FFs in multiplication model of ECC reduced the area and power consumption. We presented a novel design structure for multiplier and modulo divider with the concept of Partial Parallel Shifting (PPSM). The proposed PPSM-PCM reduced the number of logic gates that leads to high-speed operation. The comparative analysis between the proposed (PPSM-PCM) with the existing ECC architectures on the parameters of a number of logic gates, LUTs, FFs, frequency, power consumption, delay rate and latency confirmed the effectiveness.

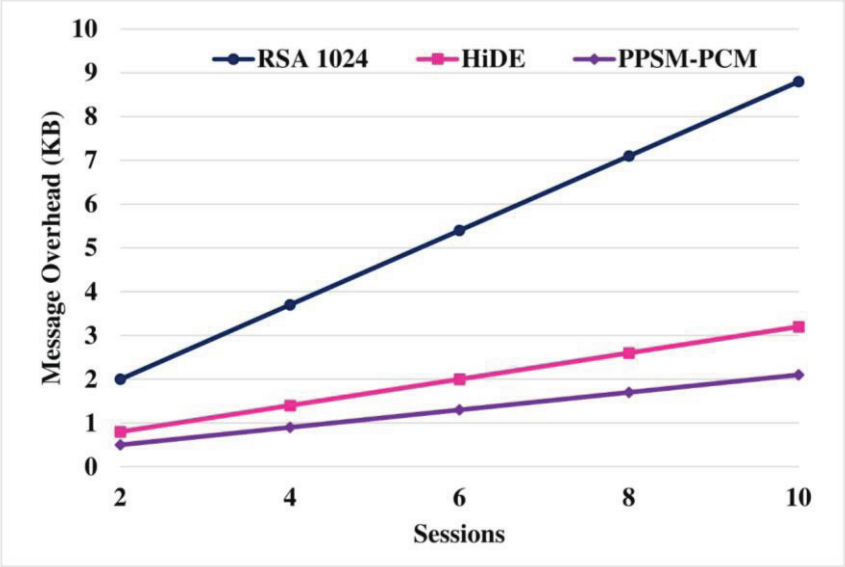

Figure 13 Message overhead vs sessions.

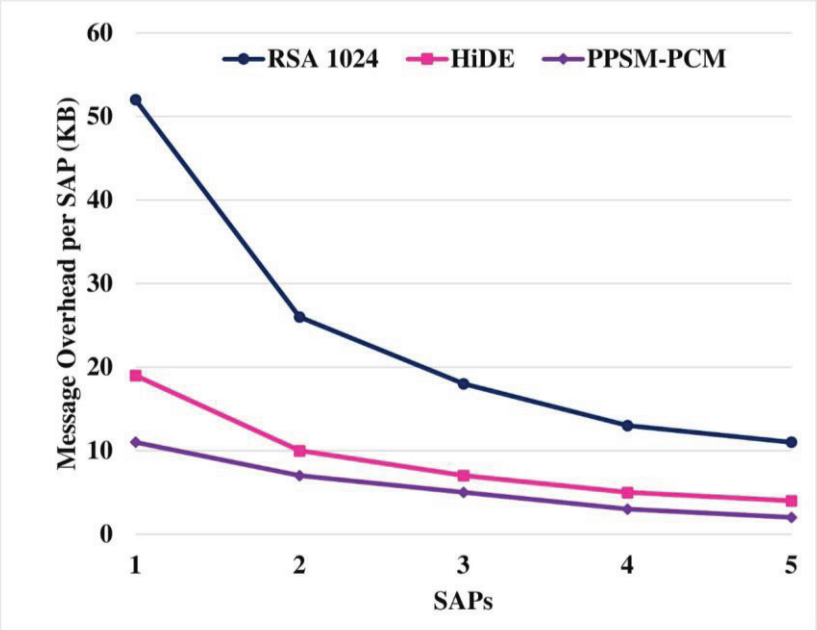

Figure 14 Message overhead vs SAP.

Table 7 Implementation Analysis.

Methods	m	Platform	Slice	Freq (MHz)	Cycles	Time (μs)
Chelton .()()()(35)	163	Virtex-4 XC4VLX200	16,209	153.9	3010	19.55
Ansari (36)	163	Virtex-4 XC4VLX200	4,080	197	4050	21
Kim (37)	163	Virtex-4 XC4VLX80	24,363	143	1446	10
Zhang (38)	163	Virtex-4 XC4VLX80	20,807	185	1428	7.7
Rebeiro (39)	163	Virtex-4 XC4VLX80	8,070	147	1429	9.7
Rebeiro(39)	163	Virtex-4 XC4VLX200	8,095	132	1429	10.7
Mahdizadeh (40)	163	Virtex-4 XC4VLX200	17,929	250	2751	9.6
Montgomery ladder (34)	163	Virtex-4 XC4VLX200	10,417	121	1091	9
ECDLP (41)	163	Virtex-4 XC4VLX200	9,300	99	-	5.75
PPSM-PCM	163	Virtex-4 XC4VLX200	2,873	252	1,141	2.95

REFERENCES

1. Aneesh R, Mohan SK, editors. Design and Analysis of High Speed, Area Optimized 32 × 32-Bit Multiply Accumulate Unit Based on Vedic Mathematics. International Journal of Engineering Research and Technology; 2014: ESRSA Publications.
2. Pöppelmann T, Güneysu T. Towards efficient arithmetic for lattice-based cryptography on reconfigurable hardware. Progress in Cryptology–LATINCRYPT 2012: Springer; 2012. p. 139–58.
3. Morales-Sandoval M, Feregrino-Uribe C, Kitsos P, Cumplido R. Area/performance trade-off analysis of an FPGA digit-serial GF (2m) Montgomery multiplier based on LFSR. Computers & Electrical Engineering. 2013;39(2):542–9.
4. I. Maglogiannis LK, K. Delakouridis, and S. Had-jiefthymiades. Enabling location privacy and medical data encryption in patient telemonitoring systems. IEEE Transactions on Informational Technology and Biomedicals. 2009;13(6):946–54.
5. Sahu SK, Pradhan M, editors. Implementation of Modular multiplication for RSA Algorithm. International Conference on Communication Systems and Network Technologies (CSNT), 2011 2011: IEEE.
6. Bhaskar R, Hegde G, Vaya P. An efficient hardware model for RSA Encryption system using Vedic mathematics. Procedia Engineering. 2012;30:124–8.
7. Adikari J, Dimitrov VS, Imbert L. Hybrid Binary-Ternary Number System for Elliptic Curve Cryptosystems. IEEE Transactions on Computers. 2011;60(2):254–65.
8. Azarderakhsh R, Reyhani-Masoleh A. Efficient FPGA Implementations of Point Multiplication on Binary Edwards and Generalized Hessian Curves Using Gaussian Normal Basis. IEEE Transactions on Very Large Scale Integration (VLSI) Systems. 2012;20(8):1453–66.
9. Rebeiro C, Roy SS, Mukhopadhyay D. Pushing the limits of high-speed GF (2 m) elliptic curve scalar multiplication on FPGAs. Cryptographic Hardware and Embedded Systems–CHES 2012: Springer; 2012. p. 494–511.
10. Sasdrich P, Güneysu T. Efficient Elliptic-Curve Cryptography using Curve25519 on reconfigurable devices. Reconfigurable Computing: Architectures, Tools, and Applications.8405:25–36.
11. Tseng CH, Wang S-H, Tsaur W-J. Hierarchical and Dynamic Elliptic Curve Cryptosystem Based Self-Certified Public Key Scheme for Medical Data Protection. IEEE Transactions on Reliability. 2015;64(3):1078–85.
12. Roy SS, Rebeiro C, Mukhopadhyay D. Theoretical Modeling of Elliptic Curve Scalar Multiplier on LUT-Based FPGAs for Area and Speed. IEEE Transactions on Very Large Scale Integration (VLSI) Systems. 2013;21(5):901–9.
13. Kumar GG, Charishma V. Design of high speed vedic multiplier using vedic mathematics techniques. International Journal of Scientific and Research Publications. 2012;2(3):1.
14. Haveliya A. FPGA implementation of a vedic convolution algorithm. International Journal of Engineering research and applications. 2012;2(1):678–884.
15. Huddar SR, Rupanagudi SR, Kalpana M, Mohan S, editors. Novel high speed vedic mathematics multiplier using compressors. International Multi-Conference on Automation, Computing, Communication, Control and Compressed Sensing (iMac4s), 2013 2013: IEEE.
16. Subudhi J, Karthick C, editors. Implementation of vedic divider on RSA cryptosystem. International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), 2015; 2015 19–20 March 2015.
17. Bathija R, Meena R, Sarkar S, Sahu R. Low Power High Speed 16x16 bit Multiplier using Vedic Mathematics. International Journal of Computer Applications. 2012;59(6):41–4.
18. Uslu B, Erdem SS. Versatile digit serial multipliers for binary extension fields. Computers & Electrical Engineering. 2015;46:29–45.
19. Sutter GD, Deschamps J, Imana JL. Efficient Elliptic Curve Point Multiplication Using Digit-Serial Binary Field Operations. IEEE Transactions onIndustrial Electronics. 2013;60(1):217–25.
20. Esmaeildoust M, Schinianakis D, Javashi H, Stouraitis T, Navi K. Efficient RNS Implementation of Elliptic Curve Point Multiplication Over <formula formulatype="inline"> <img src="/images/tex/19633.gif" alt="{\rm GF}(p)"> </formula>. IEEE Transactions on Very Large Scale Integration (VLSI) Systems. 2013;21(8):1545–9.
21. Mahdizadeh H, Masoumi M. Novel Architecture for Efficient FPGA Implementation of Elliptic Curve Cryptographic Processor Over <formula formulatype="inline"> <img src="/images/tex/21176.gif" alt="{\rm GF}(2^{163})"> </formula>. IEEE Transactions on Very Large Scale Integration (VLSI) Systems. 2013;21(12):2330–3.
22. Azarderakhsh R, Jarvinen KU, Mozaffari-Kermani M. Efficient Algorithm and Architecture for Elliptic Curve Cryptography for Extremely Constrained Secure Applications. IEEE Transactions on Circuits and Systems I: Regular Papers. 2014;61(4):1144–55.
23. Marzouqi H, Al-Qutayri M, Salah K. Review of Elliptic Curve Cryptography processor designs. Microprocessors and Microsystems. 2015;39(2):97–112.
24. Kamalakannan V, Tamilselvan S. Security Enhancement of Text Message Based on Matrix Approach Using Elliptical Curve Cryptosystem. Procedia Materials Science. 2015;10:489–96.
25. Marin L, Jara A, Gomez AS. Shifting primes: Optimizing elliptic curve cryptography for 16-bit devices without hardware multiplier. Mathematical and Computer Modelling. 2013;58(5):1155–74.
26. Anjana S, Pradeep C, Samuel P. Synthesize of High Speed Floating-point Multipliers Based on Vedic Mathematics. Procedia Computer Science. 2015;46:1294–302.
27. Chatterjee A, Sengupta I. Design of a high performance Binary Edwards Curve based processor secured against side channel analysis. VLSI Journal Integration. 2012;45(3):331–40.
28. Rajagopalan S, Amirtharajan R, Upadhyay HN, Rayappan JBB. Survey and analysis of hardware cryptographic and steganographic systems on FPGA. Journal of Applied Sciences. 2012;12(3):201.
29. Kumar KR, Boda R. FPGA Implementation of Fast Elliptic Curve Cryptography using Itoh-Tsujii algorithm. Proceedings of ICETET. 2014;29:30th.
30. Li D, Liu Y. Development of security scheme on wireless sensor network based on Elliptic Curve Cryptography. 2015.
31. Abdulrahman EA, Reyhani-Masoleh A. New Regular Radix-8 Scheme for Elliptic Curve Scalar Multiplication without Pre-Computation. IEEE Transactions on Computers. 2015;64(2):438–51.
32. Azarderakhsh R, Reyhani-Masoleh A. Parallel and High-Speed Computations of Elliptic Curve Cryptography Using Hybrid-Double Multipliers. IEEE Transactions on Parallel and Distributed Systems. 2015;26(6):1668–77.
33. Lee J-W, Chung S-C, Chang H-C, Lee C-Y. Efficient power-analysis-resistant dual-field elliptic curve cryptographic processor using heterogeneous dual-processing-element architecture. IEEE Transactions on Very Large Scale Integration (VLSI) Systems. 2014;22(1):49–61.
34. Liu S, Ju L, Cai X, Jia Z, Zhang Z, editors. High performance FPGA implementation of elliptic curve cryptography over binary fields. 2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications; 2014: IEEE.

35. Chelton WN, Benaissa M. Fast elliptic curve cryptography on FPGA. IEEE Transactions on Very Large Scale Integration (VLSI) Systems. 2008;16(2):198–205.
36. Ansari B, Hasan MA. High-performance architecture of elliptic curve scalar multiplication. IEEE Transactions on Computers. 2008;57(11):1443–53.
37. Kim CH, Kwon S, Hong CP. FPGA implementation of high performance elliptic curve cryptographic processor over GF (2163). Journal of Systems Architecture. 2008;54(10):893–900.
38. Zhang Y, Chen D, Choi Y, Chen L, Ko S-B. A high performance ECC hardware implementation with instruction-level parallelism over GF (2 163). Microprocessors and Microsystems. 2010;34(6):228–36.
39. Rebeiro C, Roy SS, Mukhopadhyay D, editors. Pushing the limits of high-speed GF (2 m) elliptic curve scalar multiplication on FPGAs. International Workshop on Cryptographic Hardware and Embedded Systems; 2012: Springer.
40. Mahdizadeh H, Masoumi M. Novel Architecture for Efficient FPGA Implementation of Elliptic Curve Cryptographic Processor Over. IEEE transactions on very large scale integration (VLSI) systems. 2013;21(12):2330–3.
41. Sghaier A, Zeghid M, Bouallegue B, Baganne A, Machhout M. Area-Time Efficient Hardware Implementation of Elliptic Curve Cryptosystem. Cryptology ePrint Archive,[web page] https://eprint. iacr. org/2015/1218. pdf, 2015.