High Throughput Multiplier Architecture for Elliptic Cryptographic Applications

Gutti Naga Swetha and Dr. Anuradha M.Sandi,

ECE department, Guru Nanak Dev Engineering College, Bidar, Karnataka, India

Abstract:

Elliptic Curve Cryptography (ECC) is one of the finest cryptographic technique of recent time due to its lower key length and satisfactory performance with different hardware structures. In this paper, a High Throughput Multiplier architecture is introduced for Elliptic Cryptographic applications based on concurrent computations. With the aid of the concurrent computing approach, the High Throughput Concurrent Computation (HTCC) technology that was just presented improves the processing speed as well as the overall efficiency of the point-multiplier architecture. Here, first and second distinct group operation of point multiplier are combined together and synthesised concurrently. The synthesis of proposed HTCC technique is performed in $Xilinx\ Virtex-5\ and\ Xilinx\ Virtex-7$ of Field-programmable gate array (FPGA) family. In terms of slices, flip flops, time delay, maximum frequency, and efficiency, the advantages of the proposed HTCC point multiplier architecture are outlined, and a comparison of these advantages with those of existing state-of-the-art point multiplier approaches is provided over $GF(2^{163})$, $GF(2^{233})$ and $GF(2^{283})$. The efficiency using proposed HTCC technique is enhanced by 30.22% and 75.31% for $Xilinx\ Virtex-5$ and by 25.13% *and* 47.75% for $Xilinx\ Virtex-7$ in comparison according to the LC design as well as the LL design, in their respective fashions. The experimental results for $Virtex-5$ and $Virtex-7$ over $GF(2^{233})$ and $GF(2^{283})$are also very satisfactory.

Keywords: *ECC, HTCC, Point Multiplication, FPGA*

1 Introduction:

In recent years, a significant change has been observed in manufacturing of electronic devices. Many electronic companies are making their devices quiet small such as micro and mini devices to keep the production cost low and enhance connectivity between appliances. Therefore, this smart electronic device is proficient enough to perform shorter computations and store their information [1]. All these data provide information about the user behaviour and environment. Hence, the information of the user behaviour and environment is the most critical data and need to be protected with high security. Hence, in this digital era, the most essential thing is information and the biggest challenge is to keep this information secure. Therefore, the demand of Cryptography is vastly enhanced to provide security for electronic devices, digital transactions and user information.

Cryptography is utilized to provide security to messages through encryption and maintain the integrity of the messages. Cryptography is widely essential for software companies, IoT devices and in services which requires privacy, trust and integrity to keep their data safe. However, in recent time, cyber-attacks on electronic devices, user information and digital transactions are widely enhanced which is a big challenge for security implementers. In modern electronic devices, data security is as essential as low power consumption and high speed performance in that device. Hence, a secure cryptosystem is highly desirable to prevent these attacks and tampering. However, the most challenging task is to maintain integrity and always keep the system safe. Hence, the most popular crypto technologies of this modern era are the RSA crypto-system [2], [3] and ECC [4, 5].

An electronic device made up of numerous heterogeneous elements whose computational capability depends upon the factors such as processing speed, storage, time delay, device area, key lengths and power consumption [1]. However, Elliptic Curve Cryptography (ECC) has much lesser key lengths, complexity and power consumption than compared to the Rivest-Shamir-Adleman (RSA) cryptosystem and

Manuscript received September 5, 2022
Manuscript revised September 20, 2022
https://doi.org/10.22937/IJCSNS.2022.22.9.54

provides similar security services as RSA. For example, RSA requires key length of 2048 bits for a particular security operation whereas ECC needs only 233 bits of key length for the same operation over a binary field [6-8]. Moreover, Elliptic Curve Cryptography (ECC) presents carry-free operations which supports hardware implementations. These above mentioned advantages makes Elliptic Curve Cryptography (ECC) more superior than Rivest-Shamir-Adleman (RSA) cryptosystem. And ECC can be very useful for embedded devices and high processing applications. Because of this, the ECC system in question is suitable for the fields of network security, health-care device operations, and smart-grid operations, all of which place a significant emphasis on preserving a high level of security [9], [10]. This is because all of these fields place a significant emphasis on preventing unauthorised access to sensitive information. This is due to the fact that each of these spheres places a substantial focus on avoiding unauthorised access to sensitive information. Because of this fact, the ECC system that is being discussed is suitable for use in the areas of network security, the operation of medical equipment, and the operation of smart grids.

Hence, many researchers have provided a significant amount of work to provide an efficient ECC system and some of the literature is presented in the following paragraph. In [11], a hard processor architecture is presented for elliptic curve cryptography (ECC) on FPGA over Galois Fields. This technique supports flexibility. However, the execution time much higher using this algorithm. In [12], a Montgomery multiplier architecture is proposed for Elliptic Curve Cryptography (ECC) for effective detection of faults. Moreover, the proposed scalar multiplication model protect against fault attacks and duplicate schemes. The Xilinx ISE FPGA serves as the platform for the implementation of this design. In the paper [13], the authors provide an efficient Vedic multiplier for Elliptic Curve Cryptography (ECC) using FPGA architecture. This multiplier is designed to deliver high speed while reducing space. Xilinx FPGA is being used for the implementation of this synthesis process. However, none of the solutions that have been presented up to this point have dealt with the issues that are brought about by using point multiplication in elliptic curve encryption. This is because none of the explanations have been comprehensive enough (ECC). This is due to the fact that these issues have not yet been brought to light. There are more issues to take into consideration, some of which include the complexity of the calculation, the length of the delay, the processing speed, and the amount of power that is used.

For this reason, the purpose of this paper is to propose a novel High Throughput Concurrent Computation (HTCC) technique that makes use of point multiplier architecture for ECC core on $Xilinx\ Virtex-5\ and\ Xilinx\ Virtex-7$ Field-programmable gate arrays (FPGA) with high efficiency. Since the suggested HTCC point-multiplier architecture is built with low latency and gives more flexibility in comparison to existing hardware devices, it is possible to simply update ECC algorithms on FPGA. Moreover, Field-programmable gate array (FPGA) are much cheaper than Application Specified Integrated Circuit (ASIC) for sample production in smaller volumes due to negligible fabrication cost. Therefore, a HTCC point-multiplier architecture is presented over binary fields $GF(2^n)$ which is synthesized in FPGA to modify algorithms of ECC cores by combining concurrent computation methods. In this article, a method known as HTCC is presented in order to improve the processing speed as well as the effectiveness of the point-multiplier architecture. The High Throughput Concurrent Computing (HTCC) methodology that has been developed is an example of a concurrent calculation method that can finish its computation in only one clock cycle. As a result, the suggested HTCC method exhibits a level of efficiency that is much greater than that of the different group operations of elliptic curve design. Hence, the proposed HTCC point- multiplier architecture shows better area-time product results and

takes lesser time for synthesis than above mentioned research works.

This article is organised in parts, which are described in the following paragraphs. In section 2, problems with the point-multiplier design on FPGA are discussed, as well as ways in which these problems may be solved with the help of the suggested HTCC approach. Section 3 describes about the mathematical background of proposed point multiplier architecture on ECC core. In section 4, experimental outcomes on FPGA and their comparison with existing algorithms is presented and section 5 concludes this paper.

2 Literature Survey:

In recent years, there has been a significant increase in the need for cryptography in a variety of disciplines, including network security, health-care devices, software firms, and smart-grid operations [9-10]. These are all areas in which a high level of security is absolutely necessary. Therefore, in order to achieve a high level of security, it is necessary to use cryptographic methods that have a minimal amount of lag time and are very efficient. Hence, there are two cryptographic techniques which are highly effective and proficient to counter the security issues such as $Rivest - Shamir - Adleman\ (RSA)$ cryptosystem [2], [3] and ECC [4, 5]. However, the utilization of ECC crypto technique is highly increased in recent years due to their lower key lengths and carry-free operations which supports hardware implementations. Therefore, Elliptic Curve Cryptography (ECC) is more superior to $Rivest - Shamir - Adleman\ (RSA)$ crypto technology. Hence, numerous researchers have proposed point multiplication architecture on FPGA which is shown in the following sections.

In [10], ECC scalar multiplication architecture is introduced to provide high speed implementation using Montgomery curves. Here, the proposed technique supports Montgomery as well as Weierstrass curves. The proposed technique reduces critical path delay and enhances performance. However, the area overhead is slightly enhanced as well. In [14], a full precision multiplier architecture is proposed for Elliptic Curve Cryptography (ECC) for enhancing processing efficiency and reduce latency. And a pipeline technique is presented to enhance frequency. This architecture is proposed on $Xilinx\ Virtex-5\ and\ Virtex-7$ FPGA. In [15], an effective hardware architecture is implemented for Elliptic Curve Cryptography (ECC) using modular multiplication to reduce area utilization. This technique is implemented on $Xilinx\ Virtex-7$ FPGA. This model performs single modular multiplication. In [16], a pipelined multiplier architecture is proposed for Elliptic Curve Cryptography (ECC) for enhancing throughput and reduce area. Moreover, this model reduces critical path delay using point multiplier architecture. This architecture is implemented on $Xilinx\ Virtex-5\ and\ Virtex-7\ FPGA$. In [17], an efficient processor is introduced for Elliptic Curve Cryptography (ECC) using point multiplication on FPGA architecture to reduce area and speed up the processing power. This technique presents point doubling operations to reduce hardware utilization. In [18], an encoding scheme is introduced for Elliptic Curve Cryptography (ECC) to set up a strong encryption mechanism. This model's primary objective is to improve the system's reliability while also increasing its capacity to deal with fault attacks and duplicate schemes. An efficient low-latency pipelined Karatsuba-ofman multiplier for elliptic curve cryptography (ECC) over Curve25519 is reported in the paper [19]. This multiplier is implemented using field-programmable gate arrays (FPGA). In order to lessen the delay caused by the network, a point multiplier design has been suggested here.

The above mentioned techniques are implemented on FPGA Xilinx based on multiplier architecture for Elliptic Curve Cryptography (ECC). Many researchers have implemented their architecture on ASIC. And Most of the researchers have focused upon modification of finite-field arithmetic model as

well as compute distinct group operations. However, it can cause higher group latency. Thus, it can cause reduction in speed of the network. Besides, the high-speed multiplier systems focuses only on area of the device. Therefore, the purpose of this paper is to propose an architecture for a High Throughput Multiplier that can be used for Elliptic Cryptographic applications. This architecture will be based on concurrent computations performed on $Xilinx\ Virtex-5\ and\ Xilinx\ Virtex-7$ Field-programmable gate arrays (FPGA) and will have a high level of efficiency. The High Throughput Concurrent Computation (HTCC) point-multiplier architecture that has been suggested is able to be implemented with low latency and gives better flexibility in comparison to other hardware devices. This indicates that ECC algorithms may be readily improved on FPGA. The HTCC point multiplier design that has been presented guarantees a trade-off between the area of the network and the speed at which it operates, which is highly recommended for cryptographic techniques.

3 Mathematical Foundations of the HTCC Approach:

This section describes about the mathematical modelling of proposed High Throughput Concurrent Computation (HTCC) technique for point multiplier architecture using Elliptic Curve Cryptography (ECC) core on FPGA. For an Elliptic Curve Cryptography (ECC), the point multiplication G can be expressed as $G = s.R$ where, s can be defined as any constant which is multiplied by a point R placed on elliptic curve to get the resultant point G [9]. The proposed HTCC point-multiplier architecture is presented over binary fields $GF(2^n)$ which is synthesized in FPGA to modify algorithms of ECC cores by combining concurrent computation methods in Cartesian coordinates. HTCC technique is a concurrent computation technique which complete its computation in just one clock cycle. Additionally, mathematical modelling of distinct group operations of elliptic curve architecture is presented and their performance is compared with the proposed HTCC technique by combining concurrent computation methods. The performance of proposed HTCC technique is highly effective than compare to distinct group operations of elliptic curve architecture. To get effective concurrent computation, the proposed HTCC technique performs trinomial and pentanomial arithmetic operations and it provides synthesis results in very less time and the area-time product results are much higher.

4 Mathematical Background of Elliptic Curve Cryptography (ECC):

Elliptic Curve Cryptography (ECC) is one of the most efficient Public Key- Encryption (PKE) method for cryptography applications due to its smaller key length for binary as well as prime fields. This paper focuses on HTCC point-multiplier architecture over binary fields $GF(2^n)$ due to its modulo-2 functional operations which provides high efficiency for hardware implementations. The most essential and critical process in Elliptic Curve Cryptography is concurrent computation of point multiplication architecture. The point multiplier architecture consists of arithmetic operations such as addition and arithmetic operations. Here, first distinct group operations represent the addition arithmetic operations and second distinct group operations represent multiplication arithmetic operations respectively. Then, the proposed HTCC technique over binary fields $GF(2^n)$ performs concurrent operations which is utilized for the execution of cryptographic system. Here, both distinct group operations of elliptic curve architecture are combined together and performed concurrently to achieve high throughput and to reduce area in hardware devices.

The combined concurrent operations for HTCC technique are performed in Affine and Cartesian coordinates. A HTCC elliptic curve point C the following equation may be used to represent data in affine coordinates: $p, q \in \mathbb{L}_2{}^n$, i.e. $R(p, q)$. However,

a HTCC elliptic curve point R Cartesian coordinates may be stated using three elements: the x, the y, and the z. $\mathbb{A}, \mathbb{B}, \mathbb{C} \in \mathbb{L}_2{}^n$, i.e. $R(\mathbb{A}, \mathbb{B}, \mathbb{C})$. Here, all the combined concurrent operations are performed in Cartesian coordinates to discard the inversion process of costly modular arithmetic functions.

A HTCC elliptic curve architecture over binary fields $GF(2^n)$ is the group of resolutions in affine coordinates for the following equation,

$$q^2 + pq = p^3 + xp^2 + y \tag{1}$$

Where, constants x, y can be expressed as $x, y \in \mathbb{L}_2{}^n$ [20] and $p, q, x, y \in GF(2^n), y \neq 0$ .Here, the no. of bits n are considered as 163 which shows that a 163-bit ECC model is presented.

Assume that R represents a point in the affine coordinate and can be defined as $R = p, q$ whereas the point R in the Cartesian coordinates can be represented as $R = \mathbb{A}, \mathbb{B}, \mathbb{C}$. Hence,

$$\mathbb{A} = p;\ \mathbb{B} = q;\ \mathbb{C} = 1. \tag{2}$$

Where, R in the Cartesian coordinates can be defined as $R = \mathbb{A}, \mathbb{B}, \mathbb{C}$ and $\mathbb{C} \neq 0$ with respect to the corresponding point R in the affine coordinates $R = p, q$ can be represented by,

$$p = \frac{\mathbb{A}}{\mathbb{C}^2};\ q = \frac{\mathbb{B}}{\mathbb{C}^3}; \tag{3}$$

By combining equation (1) and (3) in Cartesian form for the proposed HTCC elliptic curve is,

$$\mathbb{B}^2 + \mathbb{A}\mathbb{B}\mathbb{C} = \mathbb{A}^3 + x\mathbb{A}^2\mathbb{C}^2 + y\mathbb{B}^6 \tag{4}$$

Where, the point in the infinity can be expressed as $(1,1,0)$. Assume that the two points in the Cartesian coordinates in for an elliptic curve can be represented as $R = (\mathbb{A}_1, \mathbb{B}_1, \mathbb{C}_1)$ and $G = (\mathbb{A}_2, \mathbb{B}_2, \mathbb{C}_2)$. Then combined concurrent operations for HTCC elliptic curve architecture in Cartesian coordinates can be defined as,

$$K(\mathbb{A}_3, \mathbb{B}_3, \mathbb{C}_3) = 2R(\mathbb{A}_1, \mathbb{B}_1, \mathbb{C}_1) \in C(\mathbb{L}_2{}^n) \tag{5}$$

$$\mathbb{C}_3 = \mathbb{A}_1\, \mathbb{C}_1^2\,,$$

$$\mathbb{A}_3 = (\mathbb{A}_1^4 + y\mathbb{C}_1^8)\,,$$

$$\mathbb{B}_3 = \mathbb{A}_1^4\mathbb{C}_3 + (\mathbb{A}_1^2 + \mathbb{B}_1\mathbb{C}_1 + \mathbb{C}_3)\mathbb{A}_3\,;$$

$$K(\mathbb{A}_3, \mathbb{B}_3, \mathbb{C}_3) = R(\mathbb{A}_1, \mathbb{B}_1, \mathbb{C}_1) + G(\mathbb{A}_2, \mathbb{B}_2, \mathbb{C}_2) \in C(\mathbb{L}_2{}^n) \tag{6}$$

$$\mathbb{C}_3 = \mathbb{C}_1\, \mathbb{C}_2 T\,,$$

$$\mathbb{A}_3 = x\mathbb{C}_3^2 + K\,(K + \mathbb{C}_3) + T_3,$$

$$\mathbb{B}_3 = (K + \mathbb{C}_3)\mathbb{A}_3 + \mathbb{C}_1^2T^2(K\mathbb{A}_2 + \mathbb{B}_2\mathbb{C}_1T)\,;$$

Where,

$$T = (\mathbb{A}_1\mathbb{C}_2^2 + \mathbb{A}_2\mathbb{C}_1^2)\ and\ K = (\mathbb{B}_1\mathbb{C}_2^3 + \mathbb{B}_2\mathbb{C}_1^3). \tag{7}$$

Therefore, whenever the point R becomes equal to point G i.e. $R = G$ then $K = 2R$ shows the first distinct group operation of HTCC point multiplier architecture of an elliptic curve which can be represented by equation (5) and whenever $R \neq G$, then $K = R + G$ represents the second distinct group operation of HTCC point multiplier architecture of an elliptic curve which can be shown by equation (6) [21].

5. Mathematical Modelling of Proposed HTCC Point Multiplier Architecture in Cartesian Coordinates:

The most essential process of ECC core is point multiplication and it is the most costly process for an ECC core in terms of computation. On the other hand, we have developed a whole new method for Elliptic Curve Cryptography (ECC) that is based on concurrent calculations called High Throughput Concurrent Computation, or HTCC. This method uses point-multiplier architecture. The details of combined

concurrent operation and HTCC algorithm for an ECC core is explained in the following section.

The process of concurrent computation by combining first and second group operations can be shown by algorithm 1. The constant s in the following algorithm 1 is a private key and can be expressed in binary and reiterate through every bit. Usually, the first distinct group operations are simulated on each iteration whereas the second distinct group operations are simulated only when a specific bit-value of constant s becomes one i.e. bit-value of constant $s = 1$. However, HTCC technique is introduced by combining one and two group operations which provides outcomes concurrently on every cycle. Therefore, an elliptic curve needs n no. of bits to simulate the final outcome. However, every iteration requires only one clock cycle for the proposed HTCC technique.

Algorithm 1 : HTCC approach for Elliptic curve

Input: $s = \left(s_{n-1,\ldots\ldots,n_1,n_2}\right)_2, R(\mathbb{A},\mathbb{B},\mathbb{C}) \in C(\mathbb{L}_2{}^n)$

Output: $G(\mathbb{A},\ \mathbb{B},\ \mathbb{C}) = s.R(\mathbb{A},\ \mathbb{B},\ \mathbb{C})$ where, $G(\mathbb{A},\ \mathbb{B},\ \mathbb{C}) \in C(\mathbb{L}_2{}^n)$

Step 1 : $G = 0$;
Step 2: $while\ \mathrm{j} = \mathrm{n} - 1 \text{ to } 0$
$do\ G = 2G;$
$if\ s(j) =' 1'\ then\ G = G + R$
end
$end\ while$
Step 3: $Return\ \left(G(\mathbb{A},\ \mathbb{B},\ \mathbb{C})\right)$

6. Architecture of Proposed HTCC Point Multiplier:

Based on coupled concurrent computations, a new HTCC approach using point-multiplier architecture is suggested in this article. Above mentioned most of literatures have utilized distinct group operations of point multiplier for an elliptic curve. Hence, their model needs higher simulation time. The block diagram of proposed HTCC Point Multiplier is presented in the following figure 1. The architecture of proposed HTCC Point Multiplier contains pre-processor, computational block, multiplexer, LUT registers and counters etc. In this architecture, the computational block is very essential to speed up the simulation process by computing the concurrent group operations of point multiplier for an elliptic curve. The outcome of second distinct group operation using HTCC point multiplier architecture move towards output when a specific bit of 'key' tends to one. Whereas, the outcome of second distinct group operation move towards output when a specific bit of 'key' tends to zero. The outcomes of first and second group operations are placed in the LUT registers to get the resultant output. Counters are utilized to choose when these outcomes are forwarded to the following input of the HTCC technique. However, the HTCC technique requires just one clock cycle to perform concurrent computation for the resultant output of first and second distinct group operations. Therefore, for the computation of 163-bit, 233-bit and 283-bit point multiplier architecture only 163, 233 and 283 clock cycles are required respectively using HTCC technique due to its concurrent computational architecture in Cartesian coordinates.

In this article, the HTCC approach will be described for the combined concurrent calculation of the first group operation and the second group operation in point multiplier architecture. This concurrent computation helps in reduction of power consumption. As shown in figure 1, a hardware model is designed for point multiplier architecture of elliptic curve based on HTCC technique. Some observations are taken place in the entire simulation process of HTCC technique such as a uniformity is maintained in power consumption throughout the execution of concurrent computations and the extraction of 'key' information is a challenging process for anyone due to its complex hardware structure. Moreover, the timing frequency and efficiency results are very efficient and

secure compare to other state-of-art-technique using the proposed HTCC technique.

a. Group Operations Using the Newly Proposed High-Throughput Computing Technique:

Combining the first and second group operations of elliptic curve point multiplication in parallel is the basis of the new technology known as HTTC, which was developed with the goal of achieving both high throughput and low latency in Cartesian coordinates. A number of strategies, including as concurrent computing, pre-processing, and architectural balancing, among others, have been put into practise in order to achieve high speeds and efficiencies. In addition, the suggested HTCC method achieves improved results when applied to random curves as well. The combined group operations in Cartesian coordinates utilising the suggested HTCC approach are represented by the equations (5), (6), and (7) that were just described above. The suggested HTCC method is very effective and is able to successfully manage the complicated group operation of point multiplier. Seven and eleven levels, respectively, are needed in order to do the calculation for the first and second different group operations. As a result, a total of 18 levels are required in order to do separate computations for the first and second group actions. Whereas, the proposed HTCC technique utilises only 14 levels in Cartesian coordinates. This means that the proposed HTCC technique reduces the computation of group operations by four levels when compared to distinct first and second group operations. The HTCC technique accomplishes this by combining the first and second group operations of elliptic curve point multiplication. The overall number of levels in the group operations may be lowered with the assistance of the suggested HTCC approach. This indicates that the number of logic gates used can also be minimised, resulting in a noticeable gain in efficiency. The architecture of the proposed HTCC approach for elliptic curve point multiplication architecture is shown in Figure 1.

A point multiplier algorithm for $\mathbb{L}_2{}^n$ is presented with the assistance of the HTCC method that has been suggested. The entire latency of the architecture depends upon the operation of HTCC point multiplier algorithm. Therefore, the operation of this algorithm is very critical and essential to achieve low latency. Here, three unchangeable pentanomial, trinomial and pentanomial equations are utilized for 163-bit, 233- bit and 283- bit point multiplier architecture respectively which is suggested by NIST [22] such as,

$$h(p) = p^{163} + p^7 + p^6 + p^3 + 1 \quad and \quad h(p) = p^{233} + p^{74} + 1 \tag{8}$$

$$h(p) = p^{283} + p^{12} + p^7 + p^5 + 1$$

The HTCC point multiplier algorithm calculates the product of these trinomial and pentanomial equations arithmetic equations, followed by modular decrement which are shown in equation (9),

$$\mathbb{C}(p) = M(p).N(p) \bmod h(p) \tag{9}$$

Algorithm 1 : A HTCC point multiplier algorithm in $GF(2^n)$

Input: $M(p)$, $N(p) \in C(\mathbb{L}_2{}^n)$

Output: $\mathbb{C}(p) = M(p).N(p) \bmod h(p)$

Step 1 : $\mathbb{C}_N$; $R = h(p)$;

Step 2: *while* d = n − 1 to 0

do $M_N =' 0'$ *and* $M(p)$; $\mathbb{C}_N = \mathbb{C}_N.p$;

Step 3: *while* $j =$ 0 to n − 1

do $M_N(j) = M_N(j)$ *and* $N(d)$;

end while

Step 4: $\mathbb{C}_N = \mathbb{C}_N$ *xor* M_N;

Step 5: *while* 1 = 0 to n

do $R_N = R(1)$ *and* $\mathbb{C}_N(n)$;

end while

Step 6: $\mathbb{C}_N = \mathbb{C}_N$ *xor* R_N;

Step 7: *end while*

Step 8: *Return* $(\mathbb{C}(p))$

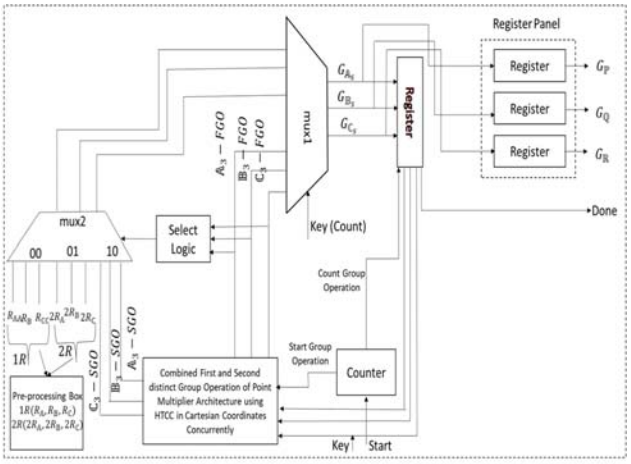

Figure 1 The suggested HTCC approach for the Elliptic curve point multiplication architecture, as seen in the architectural diagram

7. Performance Evaluation of Proposed HTCC technique for Elliptic Curve:

In this part, we will compare the performance of several state-of-the-art point multiplier approaches with the High Throughput Concurrent Computation (HTCC) methodology that uses point-multiplier architecture for Elliptic Curve Cryptography (ECC). To achieve high throughput and efficiency, the suggested HTCC approach for Elliptic Curve Cryptography (ECC) is synthesised using VHDL code and implemented in $Xilinx\ ISE$ 14.7 with integrated Xilinx Vivaldo Design suite 15. Both of these steps are carried out in $Xilinx\ ISE$ 14.7 . The synthesis of proposed HTCC technique is performed in $Xilinx\ Virtex-5\ and\ Xilinx\ Virtex-7$ of FPGA family. The synthesis findings demonstrate the superiority of the proposed HTCC point multiplier architecture in contrast with other state-of-the-art point multiplier approaches in terms of the speed, efficiency, timing, and latency of the model. The performance of the proposed HTCC techniques are compared with various point multiplier techniques such as $ECSM$ [23], BEC [24], GHC [24], $ECSMA$ [25], $DSBF$ [26], $LLSECC$ [27], $SPDSM$ [28], $HPPA$ [29], $GNBM$ [30], $BECC$ [31] , $SPFPM$ [32], $LDMPM$ [32] , LC [33] and LL [33] over163-bit ECC using $Xilinx\ Vertex-5\ and\ Xilinx\ Virtex-7$. The synthesis results are obtained by combining first and second group operation concurrently rather than distinct group operation and proposed HTCC technique provide better synthesis results compare to distinct group operations in Cartesian coordinates. Based on the synthesis results it can be observed that the efficiency of the architecture is the highest in comparison with above mentioned literatures. Furthermore, the timing, frequency and latency results are also presented for the comparison with these state-of-art-technique. The efficiency of the proposed HTCC technique can be presented by the following method,

$$Efficiency = \frac{Throughput}{Area} = n.(at)^{-1} \quad (10)$$

Where, t represents time and a shows area and n denotes number of bits.

a. Performance Evaluation over 163-bit ECC:

The point multiplier architecture for ECC curve is implemented across 163-bit ECC, 233-bit ECC, and 283 bit- ECC in several literatures. These sizes of ECC may be found in computer systems. In addition, the synthesis results are achieved across 163-bit, 233-bit, and 283-bit ECC using the suggested HTCC approach for $Xilinx\ Virtex-5\ and\ Xilinx\ Virtex-7$. This is done for the purpose of providing an impartial evaluation and comparison with other state-of-the-art techniques. The results of the synthesis reveal that generic and random curves also provide satisfactory results. Table 1 demonstrates that the suggested HTCC point multiplier architecture is preferable than the 163-bit ECC in terms of performance results and their comparison with other state-of-the-art point multiplier

approaches. Table 1 is included here for your convenience. The effectiveness of the proposed HTCC method is measured in terms of the exploitation of hardware resources such as LUC registers, slices, flip flops, time delay, maximum frequency, efficiency, and the number of clock cycles that are employed.

Table 1 compares the experimental results obtained with the proposed HTCC approach to those obtained with other state-of-the-art techniques while using $GF(2n)$ as the basis for the ECC point multiplication architecture. The suggested HTCC approach for $Xilinx\ Virtex-5\ and\ Xilinx\ Virtex-7$ has the best level of efficiency compared to any other techniques that are considered to be state-of-the-art. The timing results for $Virtex-5$ are 3.71 microseconds, whereas the timing results for $Virtex-7$ are 3.14 microseconds. In addition, the use of LUTs, flip flops, and slices is kept to a bare minimum; as a result, the architecture's size and latency may be successfully decreased. The frequency that is achieved by using the suggested HTCC method is respectable for both the $Xilinx\ Virtex-5$ and the $Virtex-7$ processors. When compared with LC architecture [33] and LL architecture [33], respectively, the efficiency of $Xilinx\ Virtex-5$ is increased by 30.22 percent when employing the suggested HTCC approach, and by 75.31 percent when compared with LL architecture [33]. Similarly, the efficiency of $Xilinx\ Virtex-7$ is improved by 25.13 percent when adopting the suggested HTCC approach, and by 47.75 percent when compared to LC architecture [33] and LL architecture [33], respectively. In contrast to the LC design [33] and the LL architecture [33], the use of slices in $Xilinx\ Virtex-5$ is decreased by 17.99 percent and 65.80 percent, respectively. Similarly, the number of slices in $Xilinx\ Virtex-7$ is reduced by 23.44 percent when compared with the LC design [33] and by 67.59 percent when compared with the LL architecture [33]. According to the findings of the synthesis, which can be seen in Table 1, the suggested HTCC approach for ECC point multiplication architecture is superior to other state-of-the-art techniques in terms of throughput, time delay, and resource consumption for 163 bits for $Xilinx\ Virtex-5\ and\ Virtex-7$.

Table 1 Evaluation of the Performance of the Proposed HTCC method in Comparison to the State-of-the-Art Techniques Using $GF(2^{163})$

Architecture	Device	LUT	FF	Slices	Freq. (MHz)	Time (us)	Efficiency
ECSM [23]	Virtex-5	10176	-	3446	16	8.6	5500
BEC [24]	Virtex-5	14235	4075	5788	264	25.3	1113
GHC [24]	Virtex-5	14235	3912	5788	267	17.7	1591
ECSMA [25]	Virtex-5	10195	-	3513	147	9.5	4854
DSBF [26]	Virtex-5	22936	-	6150	250	5.48	4837
LLSECC [27]	Virtex-5	-	-	7978	154	59.15	345
SPDSM [28]	Virtex-5	3,958	1522	1,089	296	14.06	10,630
HPPA [29]	Virtex-5	9470	4526	3041	294	4.6	11577
GNBM [30]	Virtex-5	4807	-	4815	550	94.6	358
BECC [31]	Virtex-5	-	-	5768	343	5.08	5563
SPFPM [32]	Virtex-5	16090	3090	4393	228	4.91	7557
LDMPM [32]	Virtex-5	42192	3403	11777	113	3.99	3469
LC architecture [33]	Virtex-5	9707	2034	2429	200	3.96	16936
LL architecture[33]	Virtex-5	23135	6209	5825	145	2.22	12580
HTCC	**Virtex-5**	**5211**	**2372**	**1992**	**207**	**3.71**	**22055**
BEC [24]	Virtex-7	-	-	10569	-	12.2	1264
GHC [24]	Virtex-7	-	-	6042	-	11.1	2430
SPDSM [28]	Virtex-7	4721	1886	1476	397	10.51	10507
GNBM [30]	Virtex-7	3806	-	4665	800	65	538
BECC [31]	Virtex-7	-	-	5575	437	3.97	7365
SPFPM [32]	Virtex-7	14202	3747	4150	352	3.18	12351
LDMPM [32]	Virtex-7	41090	7969	11657	159	2.83	4941
LC architecture [33]	Virtex-7	9429	2034	2435	264	3.01	22256

LL architecture [33]	Virtex-7	23011	6461	5753	214	1.5	18848
HTCC	**Virtex-7**	**3702**	**2376**	**1864**	**318**	**3.14**	**27849**

b. Performance Evaluation over 233-bit ECC:

Similarly, proposed HTCC technique is compared with state-of-art-techniques such as $ECSM$ [23], $DSBF$ [26], $LLSECC$ [27], $BECC$ [31], $HPPA$ [29], LC architecture [33], LL architecture [33] and the synthesis results are obtained over 233-bit ECC for $Xilinx\ Virtex-5\ and\ Xilinx\ Virtex-7$. Here, Table 2 shows the superiority of the proposed HTCC point multiplier architecture in terms of performance results over 233-bit ECC and their comparison with numerous state-of-art-point multiplier techniques. The performance of proposed HTCC technique is evaluated on the basis of utilization of hardware resources like LUC registers, slices, flip flops, time delay, maximum frequency, efficiency and number of clock cycles used.

Table 2 represents experimental outcomes of Proposed HTCC technique with other State-of-art-techniques using $GF(2^{233})$ for ECC point multiplication architecture. The efficiency of the proposed HTCC technique for $Xilinx\ Virtex-5\ and\ Xilinx\ Virtex-7$ is highest than any other state-of-art-techniques. The timing results are $3.79\ \mu s$ for $Virtex-5$ and $1.749\ \mu s$ for $Virtex-7$. Moreover, The utilization of LUT, flip flops and slices are minimum hence area and latency of the architecture can be reduced efficiently. The obtained frequency using the proposed HTCC technique is decent as well for both $Xilinx\ Virtex-5\ and\ Virtex-7$. The efficiency using proposed HTCC technique is enhanced by 135.80 % and 202.05 % in comparison with LC architecture [33] and LL architecture [33] respectively for $Xilinx\ Virtex-5$. Similarly, the efficiency using proposed HTCC technique is enhanced by 59.29 % and 86.64 % in comparison with LC architecture [33] and LL architecture [33] respectively for $Xilinx\ Virtex-7$. It can be observed from Table 2 that the proposed HTCC technique for ECC point multiplication architecture compare to other state-of-art-techniques is efficient in terms of throughput, time delay and resource utilization over 233-bit for $Xilinx\ Virtex-5\ and\ Virtex-7$.

Table 2 Evaluation of the Performance of the Proposed HTCC method in Comparison to the State-of-the-Art Techniques Using $GF(2^{233})$

Architecture	**Device**	**LUT**	**FF**	**Slices**	**Freq. (MHz)**	**Time (us)**	**Efficiency**
ECSM [23]	Virtex-5	18097		5644	156	12.3	3356.323
DSBF [26]	Virtex-5	22340		6487	192	19.89	1805.832
LLSECC [27]	Virtex-5			7978	154	84.19	346.8977
BECC [31]	Virtex-5			10601	360	6.84	3213.313
HPPA [29]	Virtex-5	15296	6559	4762	244	7.9	6193.547
LC architecture [33]	Virtex-5	15536	466	4001	193	5.79	10057.93
LL architecture [33]	Virtex-5	38709	9809	9729	150	3.05	7852.137
HTCC	**Virtex-5**	**6799**	**3352**	**2590**	**211**	**3.793**	**23717.74**
BECC [31]	Virtex-7			10528	497	4.91	4507.425
LC architecture [33]	Virtex-7	14861	2909	3832	266	4.19	14511.64
LL architecture [33]	Virtex-7	35976	9101	9089	221	2.07	12384.24
HTCC	**Virtex-7**	**5763**	**3118**	**5763**	**301**	**1.749**	**23116.26**

C. Performance Evaluation over 283-bit ECC:

Similarly, proposed HTCC technique is compared with state-of-art-techniques such as $DSBF$ [26], $LLSECC$ [27], $HPPA$ [29],

$LC\ architecture$ [33], $LL\ architecture$ [33]

and the synthesis results are obtained over $283-bit\ ECC$ for $Xilinx\ Virtex-5\ and\ Xilinx\ Virtex-7$. Here, Table 3 shows the superiority of the proposed HTCC

point multiplier architecture in terms of performance results over $283-bit\ ECC$ and their comparison with numerous state-of-art-point multiplier techniques. The performance of proposed HTCC technique is evaluated on the basis of utilization of hardware resources like LUC registers, slices, flip flops, time delay, maximum frequency, efficiency achieved.

Table 3 represents experimental outcomes of Proposed HTCC technique with other State-of-art-techniques using $GF(2^{283})$ for ECC point multiplication architecture. The efficiency of the proposed HTCC technique for $Xilinx\ Virtex-5\ and\ Xilinx\ Virtex-7$ is highest than any other state-of-art-techniques. The timing results are $4.42\ \mu s$ for $Virtex-5$ and $2.46\ \mu s$ for $Virtex-7$. Moreover, The utilization of LUT, flip flops and slices are minimum hence area and latency of the architecture can be reduced efficiently. The obtained frequency using the proposed HTCC technique is decent as well for both $Xilinx\ Virtex-5\ and\ Virtex-7$. The efficiency using proposed HTCC technique is enhanced by 159.97 % and 198.89 % in comparison with LC architecture [33] and LL architecture [33] respectively for $Xilinx\ Virtex-5$. Similarly, the efficiency using proposed HTCC technique is enhanced by 354.22 % in comparison with LC architecture [33] for $Xilinx\ Virtex-7$. It can be observed from Table 3 that the proposed HTCC technique for ECC point multiplication architecture compare to other state-of-art-techniques is efficient in terms of throughput, time delay and resource utilization over 283-bit for $Xilinx\ Virtex-5\ and\ Virtex-7$.

Table 3 Comparative Study of the Proposed HTCC Method with Respect to the Current State-of-the-Art Methods Using $GF(2^{283})$

Architecture	Device	LUT	FF	Slices	Freq. (MHz)	Time (us)	Efficiency
DSBF [26]	Virtex-5	25129	-	6615	188	17.78	1981.045
LLSECC [27]	Virtex-5	-	-	7978	154	102.1	286.0462
HPPA [29]	Virtex-5	20256	-	6286	213	19.9	1862.638
LC architecture [33]	Virtex-5	22476	3478	5751	167	8.12	4989.494
LL architecture [33]	Virtex-5	49313	10996	12432	128	4.31	4348.482
HTCC	**Virtex-5**	**8445**	**5229**	**4055**	**200**	**4.42**	**12973.57**
LC architecture [33]	Virtex-7	20620	3478	5417	226	6.01	7156.862
HTCC	**Virtex-7**	**5966**	**3306**	**2904**	**274**	**2.46**	**32509.79**

8. Conclusion:

The significance of cryptography is extremely essential in this digital era. Therefore, in this paper, a novel HTCC technique using point- multiplier architecture for Elliptic Curve Cryptography (ECC) core is presented. The synthesis of proposed HTCC technique is performed in $Xilinx\ Virtex-5\ and\ Xilinx\ Virtex-7$ of FPGA family over $GF(2^{163})$, $GF(2^{233})$ and $GF(2^{283})$. HTCC technique is a concurrent computation technique which complete its computation in just one clock cycle. Additionally, mathematical modelling for computation of combined concurrent group operations of elliptic curve architecture is presented and their performance is compared with various state-of-art-techniques in terms of speed, efficiency, timing and latency of the architecture. The efficiency using proposed HTCC technique is enhanced by 30.22% and 75.31% for $Xilinx\ Virtex-5$ and by 25.13% and 47.75% for $Xilinx\ Virtex-7$ in comparison with LC design and LL design respectively over $GF(2^{163})$. Additionally, the efficiency using proposed HTCC technique is enhanced by $135.80\%\ and\ 202.05\%\ for\ Xilinx\ Virtex-5$ and by $59.29\%\ and\ 86.64\%$ for $Xilinx\ Virtex-7$ in comparison with LC architecture and LL architecture respectively over $GF(2^{233})$. The utilization of LUT, flip flops and slices are minimum hence area and

latency of the architecture can be reduced effectively. The experimental results for $Virtex-5$ and $Virtex-7$ over $GF(2^{283})$ are also very satisfactory. Our Future work will be the comprehensive elliptical curve cryptography system design in firmware.

References:

[1] C. A. Lara-Nino, A. Diaz-Perez and M. Morales-Sandoval, "Elliptic Curve Lightweight Cryptography: A Survey," in IEEE Access, vol. 6, pp. 72514-72550, 2018, doi: 10.1109/ACCESS.2018.2881444.

[2] Rivest RL, Shamir A, Adleman L. A Method for Obtaining Digita Signatures and Public-key Cryptosystems. Commun. ACM. Fel 1978; 21(2): 120-126. https://doi.org/10.1145/359340.359342

[3] Kong Y, Asif S, Khan Mohammad AU. Modular multiplicatio using the core function in the residue number system. AAECC Springer Berlin Heidelberg. Jan. 2015; 27(1): 1±16.

[4] N. Koblitz, "Elliptic curve cryptosystems," Math. Comput., vo 48, no. 177, pp. 203–209, 1987.

[5] V. S. Miller, "Use of elliptic curves in cryptography," in Proc Conf. Theory Appl. Cryptograph. Techn. Santa Barbara, CA USA: Springer, 1985, pp. 417–426.

[6] Hankerson Darrel, Menezes Alfred J, Vanstone Scott. Guide t Elliptic Curve Cryptography. Springer-Verlag New York, Inc. Jan 2003.

[7] IEEE Standard Specifications for Public-Key Cryptography. IEE Standard 1363-2000. Aug. 2000; pp:1±228.

[8] NIST- National Institute of Standards and Technology, Digita Signature Standard. FIPS Publication 186-2. 2000.

[9] J. W. Bos, J. A. Halderman, N. Heninger, J. Moore, M. Naehrig and E. Wustrow, "Elliptic curve cryptography in practice," i Proc. Int. Conf. Financial Cryptogr. Data Secur. Christ Church Barbados: Springer, 2014, pp. 157–175.

[10] G. M. de Dormale and J.-J. Quisquater, "High-speed hardwar implementations of elliptic curve cryptography: A survey," J. Syst Archit., vol. 53, nos. 2–3, pp. 72–84, 2007.

[11] H. Asshidiq, A. Sasongko and Y. Kurniawan, "Implementation o ECC on Reconfigurable FPGA Using Hard Processor System 2018 International Symposium on Electronics and Smart Device (ISESD), Bandung, 2018, pp. 1-6, doi: 10.1109/ISESD.2018.8605444.

[12] M. Bedoui, B. Bouallegue, B. Hamdi and M. Machhout, "A Efficient Fault Detection Method for Elliptic Curve Scala Multiplication Montgomery Algorithm," 2019 IEEE International Conference on Design & Test of Integrated Micro & Nano Systems (DTS), Gammarth-Tunis, Tunisia, 2019, pp. 1-5, doi: 10.1109/DTSS.2019.8914743.

[13] P. Ahuja, H. Soni and K. Bhavsar, "High Performance Vedi Approach for Data Security Using Elliptic Curve Cryptography o FPGA," 2018 2nd International Conference on Trends i Electronics and Informatics (ICOEI), Tirunelveli, 2018, pp. 187 192, doi: 10.1109/ICOEI.2018.8553721

[14] J. Li, S. Zhong, Z. Li, S. Cao, J. Zhang and W. Wang, "Speed Oriented Architecture for Binary Field Point Multiplication o Elliptic Curves," in IEEE Access, vol. 7, pp. 32048-32060, 2019 doi: 10.1109/ACCESS.2019.2903170.

[15] M. M. Islam, M. S. Hossain, M. Shahjalal, M. K. Hasan and Y. M Jang, "Area-Time Efficient Hardware Implementation of Modula Multiplication for Elliptic Curve Cryptography," in IEEE Access vol. 8, pp. 73898-73906, 2020, doi: 10.1109/ACCESS.2020.2988379.

[16] Rashid, Muhammad & Imran, Malik & Jafri, Atif & Kashi Muhammad. (2019). A Throughput/Area Optimized Pipeline Architecture for Elliptic Curve Crypto Processor. IET Computer & Digital Techniques. 13. 10.1049/iet-cdt.2018.5056.

[17] M. M. Islam, M. S. Hossain, M. K. Hasan, M. Shahjalal and Y. M Jang, "FPGA Implementation of High-Speed Area-Efficien Processor for Elliptic Curve Point Multiplication Over Prim Field," in IEEE Access, vol. 7, pp. 178811-178826, 2019, doi: 10.1109/ACCESS.2019.2958491.

[18] H. N. Almajed and A. S. Almogren, "SE-Enc: A Secure an Efficient Encoding Scheme Using Elliptic Curve Cryptography in IEEE Access, vol. 7, pp. 175865-175878, 2019, doi: 10.1109/ACCESS.2019.2957943.

[19] R. Salarifard and S. Bayat-Sarmadi, "An Efficient Low-Latenc Point-Multiplication Over Curve25519," in IEEE Transactions o Circuits and Systems I: Regular Papers, vol. 66, no. 10, pp. 3854 3862, Oct. 2019, doi: 10.1109/TCSI.2019.2914247.

[20] P. Choi, M. Lee, J. Kim and D. K. Kim, "Low-Complexity Ellipti Curve Cryptography Processor Based on Configurable Partia Modular Reduction Over NIST Prime Fields," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 6 no. 11, pp. 1703-1707, Nov. 2018, doi: 10.1109/TCSII.2017.2756680.

[21] Orlando G and Paar C. A High-Performance Reconfigurabl Elliptic Curve Processor for GF(2m). In:Proc. CHESS. 2000. pp 41±56.

[22] D. F. P. Gallagher and C. Director, "FIPS PUB 186-3 federa information processing standards publication digital signatur standard (DSS)," Federal Inf. Process. Standards Publication 2009.

[23] C. Rebeiro, S. S. Roy, and D. Mukhopadhyay, "Pushing the limi of high-speed *GF*(2*m*) elliptic curve scalar multiplication o FPGAs," in *Proc. Int. Workshop CHES*, 2012, pp. 494–511.

[24] R. Azarderakhsh and A. Reyhani-Masoleh, "Efficient FPGA implementations of point multiplication on binary Edwards and generalized Hessian curves using Gaussian normal basis," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 20, no. 8, pp. 1453–1466, Aug. 2012.

[25] S. S. Roy, C. Rebeiro, and D. Mukhopadhyay, "Theoretical modelling 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.

[26] G. D. Sutter, J. Deschamps, and J. L. Imaña, "Efficient elliptic curve point multiplication using digit-serial binary field operations," *IEEE Trans. Ind. Electron.*, vol. 60, no. 1, pp. 217–225, Jan. 2013.

[27] K. C. C. Loi, S. An, and S.-B. Ko, "FPGA implementation of low latency scalable elliptic curve cryptosystem processor in *GF(2m)*," in *Proc. ISCAS*, Jun. 2014, pp. 822–825.

[28] Z.-U.-A. Khan and M. Benaissa, "Throughput/area-efficient ECC processor using montgomery point multiplication on FPGA," *IEEE Trans. Circuits Syst. II, Express Briefs*, vol. 62, no. 11, pp. 1078–1082, Nov. 2016.

[29] L. Li and S. Li, "High-performance pipelined architecture of elliptic curve scalar multiplication over GF(2*m*)," *IEEE Trans.*

Very Large Scale Integr. (VLSI) Syst., vol. 24, no. 4, pp. 1223–1232, Apr. 2016.

[30] T. T. Nguyen and H. Lee, "Efficient algorithm and architecture for elliptic curve cryptographic processor," *J. Semicond. Technol. Sci.*, vol. 16, no. 1, pp. 118–125, 2016.

[31] B. Rashidi, R. R. Farashahi, and S. M. Sayedi, "High-speed hardware architecture of scalar multiplication for binary elliptic curve cryptosystems," in *Proc. IEEE Conf. Inf. Secur. Cryptol. (ISCISC)*, Sep. 2014, pp. 15–20.

[32] Z. U. Khan and M. Benaissa, "High-speed and low-latency ECC processor implementation over GF(2*m*) on FPGA," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 1, pp. 165–176, Jan. 2017.

[33] R. Salarifard, S. Bayat-Sarmadi and H. Mosanaei-Boorani, "A Low-Latency and Low-Complexity Point-Multiplication in ECC," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 9, pp. 2869-2877, Sept. 2018, doi: 10.1109/TCSI.2018.2801118.

Ms. G. Naga Swetha, Research Scholar, Department of E&CE, Guru Nanak Dev Engineering College, Affiliated to Visvesvaraya Technological University (VTU), Public university in Belgaum, Karnataka. She is presently working as Asst Professor in Dept. of ECE, Madanapalle Institute of Technology & Science, Madanapalle, Andhra Pradesh. She has 15 years of teaching experience and she Completed her M.Tech in Embedded Systes in Jawaharlal Nehru Technological University, Anantapuramu. She did B.Tech – EIE in Jawaharlal Nehru Technological University, Anantapuramu. She is doing the present research work on Cryptography and VLSI design. She has 5 reputed international journals and 6 International Conferences.

Dr. Anuradha M. Sandi, Professor & R&D Coordinator, Department of E&CE, Guru Nanak Dev Engineering College, Affiliated to Visvesvaraya Technological University (VTU), Public university in Belgaum, Karnataka. She done her research in the Area of Specialization - Micro and Nano Characterization. She is a recognized supervisor under Visvesvaraya Technological University (VTU), Public university in Belgaum. She has 25 years of teaching experience in E&CE. She completed her PhD in Gulbarga University, Gulbarga and M.Tech from VTU Belgaum. She published 32 journals in reputed National & International level and 21 Conference papers in reputed National & International level.