

# Montgomery Multiplication with a Serial to Parallel Multiplier

Posted on October 9, 2026

View this essay online:

<https://academic-master.com/montgomery-multiplication-serial-to-parallel-multiplier/>

## ABSTRACT

Elliptic Curve Cryptography (ECC) has become a strong and effective method for secure data protection and communication. Montgomery multiplication is a foundation and key operation for point multiplication in ECC. In Elliptic Curve Cryptography (ECC), Montgomery multiplication is a commonly used method for expediting modular arithmetic operations. However, typical Montgomery multipliers have high latency and large area requirements, which limits their application in contexts with limited resources. In this study, a novel design for a high-speed, small-area Montgomery multiplication method using a Serial to Parallel Multiplier (SPM) is proposed and put into practise for usage in ECC applications. The main goal of this research is to maximise computational Montgomery multiplication efficiency while minimising hardware resource footprint. To accomplish this, the study uses a novel strategy that takes advantage of the parallelism that SPM naturally provides while optimising them specifically for ECC operations. This device is specifically designed to meet the specific needs of ECC, assuring compatibility with various security levels and curve parameter ranges. Modern Very Large-Scale Integration (VLSI) techniques are used in the implementation to strike a balance between fast processing and small silicon area requirements. Additionally, this research investigates the possibility of hardware acceleration of ECC processes, which is increasingly important in contexts with limited resources, such as embedded systems and IoT devices. In comparison to previous techniques, the proposed SPM-based Montgomery multiplier considerably reduces latency by performing one modular multiplication in a single clock cycle. The suggested SPM-based Montgomery multiplier can also minimise the area by using the same hardware resources for addition and multiplication operations. As a result, this study makes a contribution to the field by developing a low-area, high-performance Montgomery multiplication algorithm that meets the computational needs of ECC for secure communication and [data security](#).

Keywords: Elliptic Curve Cryptography (ECC), Montgomery multiplication, Very Large-Scale Integration (VLSI), Serial to Parallel Multiplier (SPM), Delay, Data Security.

# INTRODUCTION

Elliptic curve cryptography (ECC) is a public key encryption method founded on elliptic curve theory. It can be used to produce **cryptographic keys** more efficiently, quickly, and in smaller sizes. ECC is a substitute for the Rivest-Shamir-Adleman (RSA) cryptographic algorithm and is most frequently used for one-way email, data, and software encryption as well as digital signatures in cryptocurrencies like Bitcoin and Ethereum [1]. An elliptic curve is not an oval shape, but rather a looping line that intersects two axes, or lines on a graph that show where a point is located [2]. Along the graph's x-axis, the curve is perfectly symmetrical, or mirrored. Systems that use public key cryptography, like ECC, combine two different keys through a mathematical procedure, which is then used to encrypt and decrypt data [3]. One is a private key that is only known by the sender and recipient of the data, while the other is a public key that is known to everyone. Instead of using the conventional approach of creating keys by multiplying big prime numbers, ECC creates keys using the characteristics of an elliptic curve equation [4] [5]. The process of repeatedly adding a point along an elliptic curve to itself is known as elliptic curve scalar multiplication. A key component of ECC, which is commonly used for digital signatures and secure communications is elliptic curve multiplication [6]. ECC is favoured among all public key algorithms because it offers higher security per key bit. However, ECC calculation requires a lot of mathematics and adds a lot of time for applications that require speed. The Montgomery multiplication is used in ECC to improve the speed of operation for cryptography [7].

Montgomery multiplication is a technique for quick modular multiplication that is frequently employed in number theory and cryptography. The American mathematician Peter L. Montgomery introduced this method in the year 1985 [8]. This strategy is used in ECC to speed up the computation of scalar multiplication, which is the process of repeatedly adding a point along an elliptic curve to itself. This method reduces the number of partial products and enhances the speed of multiplication process. The effectiveness of Montgomery multiplication in cryptography is closely related to safety of cryptographic methods [9] [10]. Fast modular multiplication is a crucial part of strong cryptographic solutions because it lessens the vulnerability to side-channel attacks that take advantage of time or power consumption patterns. This multiplication is used to create a cryptographic key. The time needed to generate strong cryptographic keys directly depends on the speed and effectiveness of Montgomery multiplication [11]. There are various multiplier circuits such as binary, array, booth, serial to parallel, Wallace tree and so many multipliers used in digital circuit. Among aforementioned multiplier circuits the serial to parallel multiplier is one of the efficient multipliers to improve the speed and reduce delay of multiplication process in circuits [12].

A serial-to-parallel multiplier is a kind of multiplier that multiplies two binary values by adding their partial products simultaneously [13]. It adds the partial products serially and is faster than a serial-

serial multiplier. This multiplier multiplies two inputs in a serial manner and produces the final output in parallel format which means it receives the inputs one by one and processes them, then stores and generates all outputs simultaneously and hence the delay of the operation is reduced, increasing the operating speed. Also, it reduces the area and power consumption of the circuit. Power, delay and speed are important factors in determining system performance. This multiplier may be more suitable for many applications since it has less time delay. Montgomery multiplication used in ECC has some drawbacks. The computation takes longer and is more complicated since there are additional conversions to and from the Montgomery form [14]. As a result, it is ineffective for reductions of a single modular unit and only profitable for chains of modular operations. Due to the requirement to limit the final result to the  $[0, N]$  range, there may be security vulnerabilities introduced [15]. Attackers may be able to use timing variations or information about the operands or modulus that is revealed as a result. To overcome these limitations in existing method, a novelty of serial to parallel multiplier is introduced in Montgomery multiplication in this study. This introduced approach reduces the latency and increases the speed.

In this paper, the high speed and area efficient Montgomery multiplication with serial to parallel multiplier is proposed in ECC. This strategy increases the operating speed while reducing delay. The main objectives of this paper are as follows:

The group operation's data path is optimized, the number of stages in the process is decreased, and the system's performance is increased.

The recommended Montgomery method offers the benefit of lowering latency in comparison to many other algorithms.

In this work, a mathematical method for achieving the suggested multiplier structure is presented to reduce time delay.

The advantage of the suggested technique is that it allows for the right scheduling and projection functions to be chosen in order to derive the ideal architecture for ECC.

The layout of this essay is as follows. The related literature of Montgomery multiplication and their problem statement is listed in Section 2. The detailed architecture of suggested SPM with Montgomery is shown in Section 3. The analysis and outcomes of the experiment are covered in Section 5. The work of this article is summarised in Section 6.

# LITERATURE REVIEW

In 2019, Li, et. al [16] have offered a unique architecture for point multiplication in elliptic curve encryption with an emphasis on speed optimisation. The suggested design consists of a modified Montgomery Ladder algorithm, a balanced full-precision multiplier, and a novel modular inversion architecture. In comparison to all other efforts, the suggested design has the lowest latency, which makes it extremely effective for point multiplication in elliptic curve encryption. However, most designs are less useful in real-world applications than lab trials since they can't sustain high frequency in bigger fields. Future multiple-field point multiplication can be supported by the architecture by being updated to a reconfigurable design.

In 2020, Mehrabi, et. al [17] have suggested an RNS-based ECC core hardware for two types of elliptic curves, namely short Weierstraß and twisted Edwards curves, which are commonly utilised in security modules for secure communications in IoT applications. The effectiveness and speed of ECC point multiplication are further improved by the addition of high-speed SOR units and parallel computation in the RNS point arithmetic. In the study, an RNS-based ECC core hardware is proposed for secure communications in Internet of Things (IoT) applications, which can offer improved security for sensitive data. In addition to short Weierstraß and twisted Edwards curves, other families of elliptic curves, such as Montgomery curves or Hessian curves, may also benefit from the use of RNS-based ECC core technology.

In 2020, Rashid, et. al [18] have provided a high-speed hardware architecture for binary fields over GF ( $2^{163}$ ) and GF ( $2^{571}$ ) point multiplication (PM) computation of elliptic-curve encryption. The suggested design incorporates parallel modular multipliers, a serially connected adder, and two serially connected squarer units to decrease the computing time for PM calculation. For cryptographic applications, the study was suggested a high-speed elliptic-curve point multiplication (PM) architecture. provide point multiplication (PM) calculation for elliptic-curve encryption an effective, high-speed hardware design that reduces computation time and overall clock cycles. The potential of the suggested high-speed architecture for point multiplication (PM) computation in different cryptographic algorithms and field sizes may be investigated through further study.

In 2020, Abd-Elkader, et. al [19] have introduced the Montgomery Modular Multiplier (MMM) architecture's FPGA-based optimised implementation. The suggested design's innovative architecture increased both the maximum frequency and the area occupied on the intended FPGA. For use with generic GF(p), the suggested architecture is implemented as a modular multiplier for lightweight ECC. Throughput of this research has been improved and efficiency of implementation also had greatest performance.

In 2021, Zhao, et. al [20] focussed on enhancing the security and effectiveness of the Physical Unclonable Function (PUF) response encryption in Internet of Things (IoT) devices by utilising the Elliptic Curve Cryptography (ECC) technique with various key lengths. The point multiplication circuit based on a binary field with adjustable key lengths for ECC encryption is described in detail in this work. The Virtex-6 FPGA from Xilinx is used to implement the circuit. The circuit provides versatility in encryption operations by allowing the use of four key lengths (233, 283, 409, and 571) to suit the needs of PUF response encryption. There has been little exploration of potential trade-offs or difficulties posed by adjustable key lengths and how they may affect system performance as a whole. The point multiplication circuit should be further optimised to increase its effectiveness and use less resources.

In 2021, Ahmadpour and Jaberipur [21] have suggested a method for quick 16-bit residue channels for fast hardware Montgomery modular multiplication (MMM) implementation in residue number systems (RNS) for ultra-wide key-lengths up to 8192 bits. The suggested delta-independent modulo-(2-delta) adders, multipliers, and a unique mechanism for choosing moduli for base extension form the basis of the scheme. The differential power-analysis (DPA) attacks are prevented by the suggested approach for moduli selection for base extension, hence boosting the security of the cryptosystems. This approach required large storage area.

In 2022, Ibrahim, et. al [22] have suggested a one-dimensional bit-parallel semi-systolic multiplier for the Montgomery multiplication method, a key operation in cryptographic protocols, with low-area, low-energy, and high-speed. The suggested multiplier structure is different from earlier two-dimensional parallel systolic designs in that it has an area complexity of order  $O(m)$  rather than  $O(m^2)$ . The suggested multiplier design reduces space by a small amount while significantly reducing latency. This research recommended the upcoming project, the suggested multiplier design into the ECC crypto-processor were integrate to calculate the system's total space, time, and energy savings.

In 2022, Charishmmaa, et. al [23] have suggested a redundant signed digit representation-based exportable application-specific instruction-set ECC processor. The processor had a high throughput modular divider and an effective modular adder in addition to significant pipelining. The study suggested a redundant signed digit representation-based, heavily pipelined, exportable application-specific instruction-set ECC processor. The processor's significant pipelining can improve throughput and performance, enabling quicker execution of cryptographic procedures. creating a novel reversible gate and incorporating reversible logic into a comprehensive quantum processor capable of processing at extremely high speeds and using extremely little power was suggested for future study.

In 2022, Awaludin, et. al [24] have recommended the quickest implementation of ECC point

multiplication over Curve448 as a high-performance architecture for ECC. By fully parallelizing and pipelining, it proposes a unique variation of the Karatsuba formula for asymmetric digit multiplier, lowering the number of DSPs needed while maintaining performance. The unique Karatsuba formula for asymmetric digit multiplier has some drawbacks that are not investigated, such as its application to other cryptographic methods or its performance with various operand widths. The suggested architecture's viability and appropriateness for resource-constrained contexts would be shown by looking into the energy efficiency and power consumption of the system in future research.

In 2023, Jain and Bansod [25] have suggested an improved Montgomery algorithm for the efficient implementation of modular multiplication in cryptography in the framework of RSA method. Verilog is used to implement the method, while Vivado is used to simulate it. Utilising an FPGA NESYS 4 DDR hardware board, physical testing is conducted. In this work, an improved Montgomery method is proposed for the effective implementation of modular multiplication in cryptography. In this work, an improved Montgomery method is proposed for the effective implementation of modular multiplication in cryptography.

## Problem Statement

Existing Montgomery multiplication techniques using various multipliers for ECC are discussed in this section. Table 2 depicts the comparison of contribution, methodology used in article, benefits, and drawbacks of various aforementioned research.

Table 2. Comparison of existing methods

| Year & Author                      | Method used                               | Aim                                                                                                        | Advantages                                                                      | Limitations                                                                                  |
|------------------------------------|-------------------------------------------|------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
| [16]<br>2019,<br>Li,<br>et. al     | Balanced Full-Precision Multiplier (BMUL) | To decrease latency and boost performance, a speed-oriented design for PM was suggested.                   | Lowest latency<br>Highly efficient.<br>Higher security level.                   | This design cannot maintain high frequency.                                                  |
| [17]<br>2020<br>Mehrabi,<br>et. al | Residue Number Systems (RNS)              | To propose an RNS hardware architecture supporting fast point-addition, doubling, and tripling operations. | Enhanced security for sensitive data.<br>Improved efficiency<br>Increases speed | This method is not suitable for bigger elliptic curves or trickier cryptographic procedures. |

|                                         |                                           |                                                                                                                                                       |                                                                                            |                                                                                                                                                |
|-----------------------------------------|-------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| [18]<br>2020,<br>Rashid,<br>et. al      | Elliptic-Curve<br>Point<br>Multiplication | To offer an effective high-speed architecture for PM computation of EC encryption employing binary fields over GF ( $2^{163}$ ) and GF ( $2^{571}$ ). | Provide Efficient high-speed hardware.<br>Reduced computational time.                      | The suggested architecture's limits in terms of its suitability for a range of high-speed applications, including banking and mobile security. |
| [19]<br>2020<br>Abd-Elkader, et. al     | Modular<br>Multiplier                     | To increase the frequency and decrease the area of the MMM algorithm.                                                                                 | Increased throughput.<br>It occupies less area.<br>Great implementation efficiency         | The implemented design is only applicable for lightweight ECC.                                                                                 |
| [20]<br>2021<br>Zhao,<br>et. al         | Point<br>Multiplication<br>(PM)           | To address the needs of system compatibility and increased system flexibility.                                                                        | Security enhancement.<br>Improving efficiency for encryption.<br>Reconfigurable key length | Reconfigurable key lengths may impact system performance.                                                                                      |
| [21]<br>2021<br>Ahmadpour and Jaberipur | free-δ RNS<br>MMM                         | To present a 16-bit residue channel-based hardware implementation of MMM in RNS for ultra-wide key.                                                   | higher security in public-key cryptosystems.<br>Efficient arithmetic operation.            | Huge storage is required.                                                                                                                      |
| [22]<br>2022<br>Ibrahim,<br>et. al      | Semi Systolic<br>Multiplier               | To design a 1-D bit-parallel semi-systolic multiplier for Montgomery multiplication that is low-area, energy, and high speed.                         | Area reduced<br>Less delay.<br>Improved efficiency.                                        | This approach is applicable for limited resources only.                                                                                        |
| [23]<br>2022<br>Charishmmaa,<br>et. al  | Reversible<br>Technique                   | To overcome the traditional speed-power trade-off and move closer to the realization of quantum computing systems                                     | Improved security.<br>Efficiency enhancement.                                              | It might have difficult to implement in hardware.                                                                                              |

|                                       |               |                                                                                                               |                                                     |                                                      |
|---------------------------------------|---------------|---------------------------------------------------------------------------------------------------------------|-----------------------------------------------------|------------------------------------------------------|
| [24]<br>2022,<br>Awaludin,<br>et. al  | Karatsuba     | To offer a high-performance ECC over Curve448 architecture.                                                   | Reduces the number of required DSP.                 | This approach had longer addition tree               |
| [25]<br>2023<br>Jain<br>and<br>Bansod | RSA algorithm | To provide an improved Montgomery method for implementing modular multiplication effectively in cryptography. | Less power consumption. Improvement in performance. | This method may not suitable for larger input sizes. |

## PROPOSED METHODOLOGY

This section of the paper will discuss the recommended strategy. The truth is that rather simple modular arithmetic is the foundation upon which many complex encryption methods are constructed. The only operations that can be carried out on integers are addition, subtraction, multiplication, and division since integers are the only kind of numbers that can be utilised in modular arithmetic. All operations are only carried out in reference to a positive integer, also referred to as the modulus, in modular arithmetic. Montgomery multiplication is a technique for quickly conducting modular multiplication for public key cryptography techniques in ECC. This multiplication can be implemented in both software and hardware with less delay, power, and area. The need for conversions to and from the Montgomery space, which entail  $R$ 's multiplications and reductions, is one of the factors contributing to the longer Montgomery multiplication times. Time delay is the main factor to decide the system performance. To improve the efficiency of device, the time delay should be reduced. Therefore, the Serial to Parallel Multiplier is introduced instead of traditional multiplier in Montgomery multiplication. In SPM, the input is given to multiplier one by one serially and the multiplier unit processing it then generates the result in parallel mode. Hence the delay of the multiplication unit will be reduced and obviously power consumption also automatically reduced since it has less time delay. Figure 1 illustrates the basic structure of introduced method.

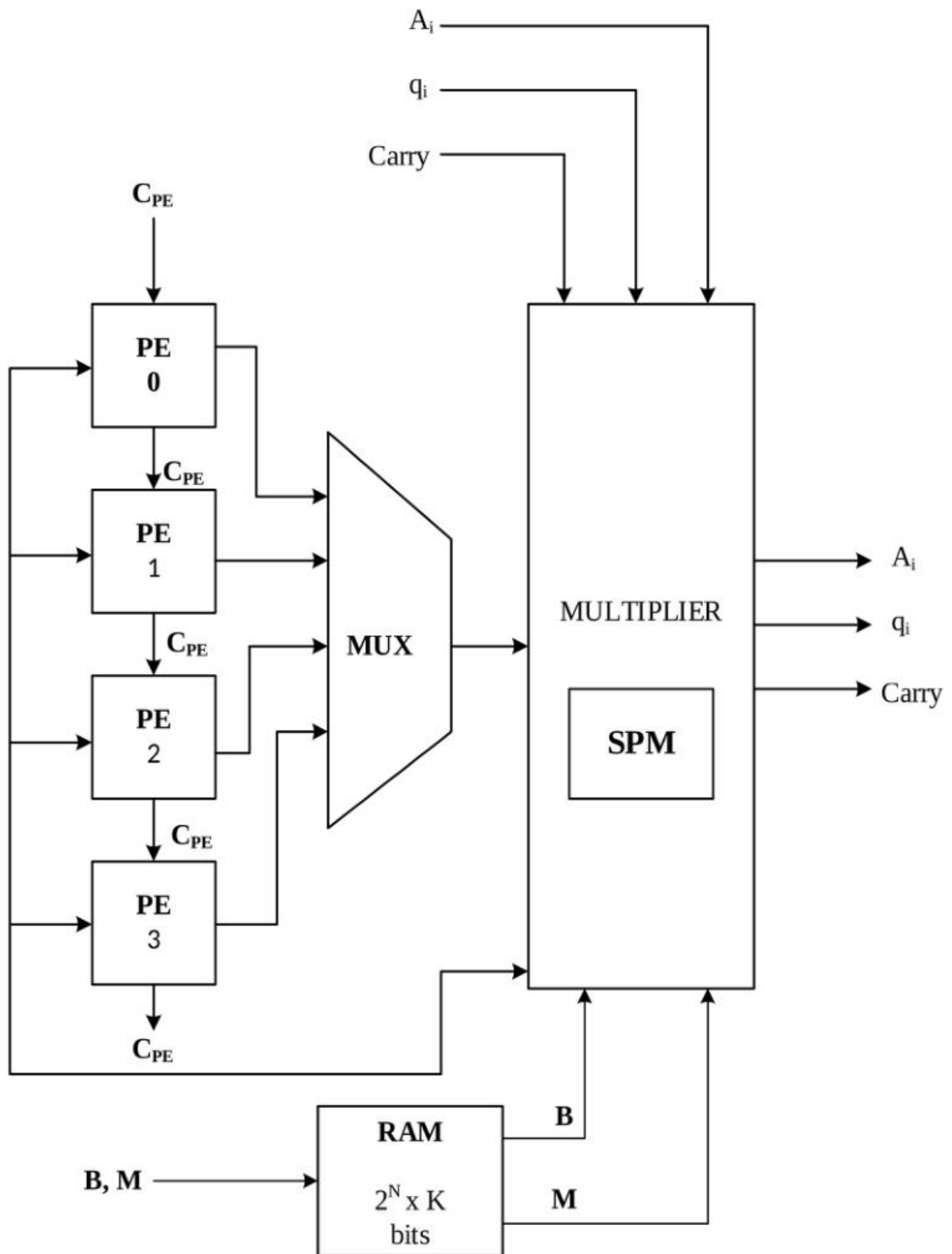


Figure 1. Block diagram of proposed Montgomery SPM

## Montgomery Multiplication for ECC

An effective and quick modular multiplication may significantly speed up the total elliptic curve scalar multiplication process. Multiplication is a crucial operation in the ECC algorithm. Through a straightforward shift, Montgomery multiplication converts division by  $p$  into division by a power of 2, performing modular multiplication. In order to do all arithmetic operations in the Montgomery domain and subsequently convert back to the ordinary domain, the inputs must first be converted into the Montgomery domain. The modular multiplication of two numbers  $x$  and  $y$  modulo  $N$  (referred to as  $N$ -residues) using Montgomery's approach is efficient and may be expressed as follows:

$$\text{Mont}(x,y) = xy R^{-1} \bmod N \quad (1)$$

The Montgomery multiplier for ECC uses registers, a systolic array, a comparator and a counter as displayed in Figure 2. This architecture's inputs were taken as  $X$ ,  $Y$ , and  $M$  1-bit inputs at first, expanding to  $n$  bits later; **START** is the controller's directive for taking the system out of standby **DONE** (signify end of operation) carries state and outputs. Result outputs a single or  $n$ -bit result.

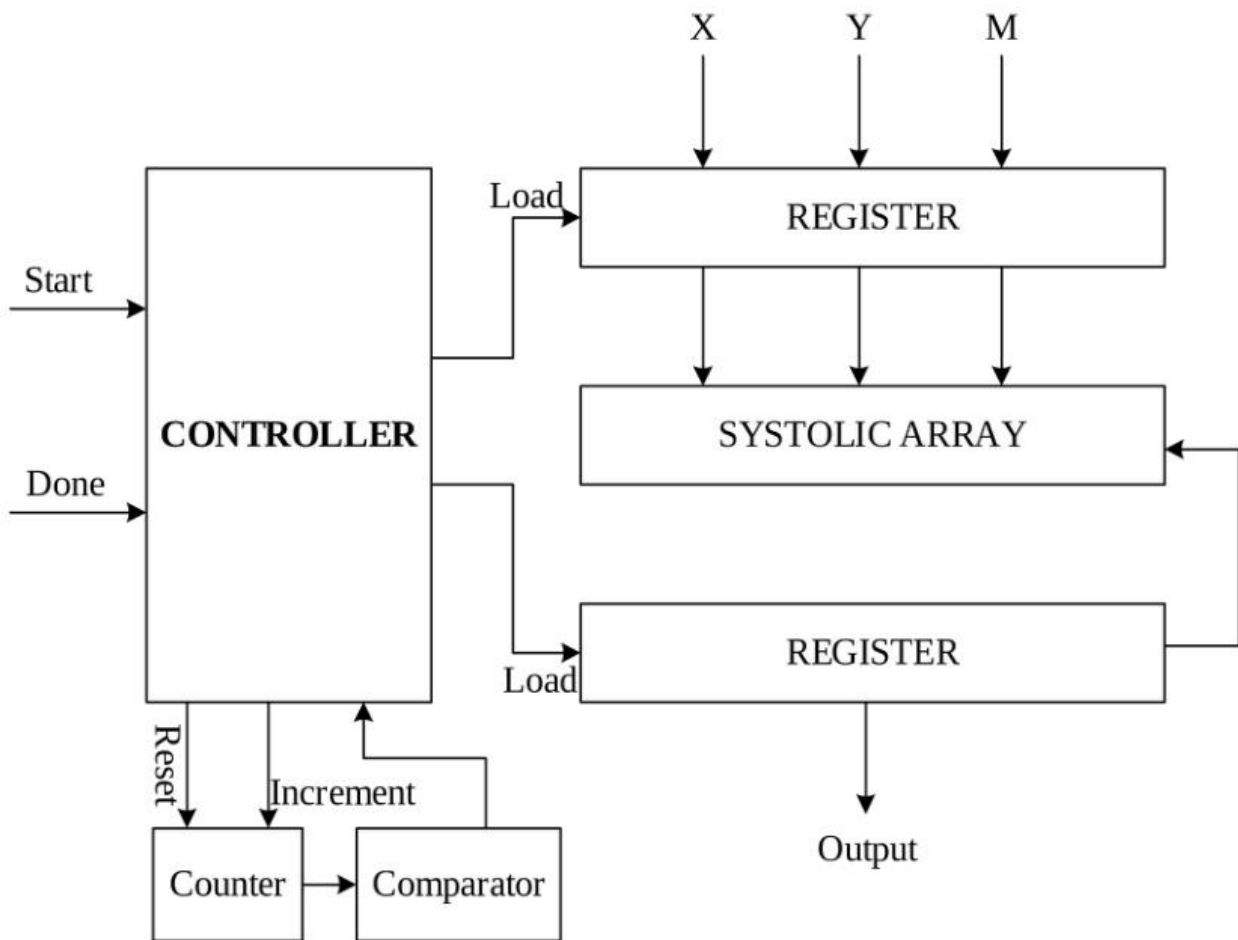


Figure 2. Architecture of Montgomery multiplier for ECC

The system stays in an IDLE state until the START command is applied to controller. The input signals X, Y, and M are pushed into register after giving START instruction. The register and counter are reset when START is interfaced, and the output value of the systolic array cells is then carried out in register. The controller increases the counter by 1 to enter the next state while it is in current one. When the counter reaches the value of  $2(l+1)$ , the comparator generates the Count-end signal, which causes the controller to enter the OUT state. The value for the RESULT output is simultaneously sent to the register. At the very end of the process, the acknowledgment signal DONE is set. The procedure is complete when the OUT state is met because RESULT is created from the register and signal DONE is set.

In the Montgomery technique, divisions in traditional reduction algorithms are swapped out for less expensive operations like shift and addition. The technique works well for calculations like modular exponentiation but is inefficient for a single modular multiplication. Algorithm 1 describes Montgomery modular multiplication, where  $n$  is the  $k$ -bit module,  $r=2^k$ , and  $(rxr^{-1} - nxn') = 1$  may be

pre-computed. A 2k-bit addition, a k-bit conditional subtraction, and three k-bit multiplications are all included in the Montgomery multiplication. The algorithm of this multiplication is shown below:

#### Algorithm 1 Montgomery Multiplication

Input:

$X = (X_{n-1}, X_{n-1}, \dots, X_0)$ ,  $Y = (Y_{n-1}, Y_{n-1}, \dots, Y_0)$ ,  $M = (M_{n-1}, M_{n-1}, \dots, M_0)$

Output:

$q = (q_{n-1}, q_{n-1}, \dots, q_0)$

Initialization:

$A^0 = (a_{m-1}^0, a_{m-2}^0), \dots, a_0^0$

$B^0 = (b_0^0, b_1^0), \dots, b_{m-1}^0, b_m^0$

1: for  $1 \leq i \leq ((m+1))/2$  do

2:  $a_{-1}^{i-1} = 0$

3:  $b_m^{i-1} = 0$

4: for  $0 \leq j \leq m-1$  do

5:  $a_{m-1-j}^i = a_{m-2-j}^{i-1} + a_{m-1}^{i-1} f_{m-1-j} + Y_{m-i} X_{m-1-j}$

6:  $b_j^i = b_{j+1}^{i-1} + b_0^{i-1} f_{i+1} + Y_{i-1} X_j$

7: end for

8: end for

9: for  $0 \leq j \leq m-1$  do

10:  $q_j = a_{m-1-j}^{(m+1)/2} + b_j^{(m+1)/2}$

11: end for

## Serial to Parallel Multiplier

A type of multiplier known as a serial-to-parallel multiplier is capable of quickly and effectively multiplying two binary values. It operates by multiplying the bits of the partial products in each column and shifting the results one at a time into a register. In digital signal processing and computer architecture, a serial to parallel multiplier is a digital circuit or algorithmic approach that effectively multiplies two integers by splitting the multiplication work into smaller, easier parts. A serial to parallel multiplier handles the multiplication bit by bit in a sequential manner as opposed to ordinary parallel multipliers, which execute simultaneous computations on all bits of the operands. The least significant bits of the operands are multiplied first, and the results are then accumulated. Before processing all of the operands' bits, it shifts and multiplies the subsequent set of bits. The Least Significant Bit (LSB) of the product are available in serial while the Most Significant Bit (MSB) bits are available in parallel in the SPM. The MSB can be timed into a register at the end of  $(n + nl)$  clocks using the same clock source if  $nl$  is the number of clock intervals needed to cover the parallel adder delay. The SPM type multiplier is mainly used to reduce latency in digital circuit.

In SPM, the inputs are applied serial manner. The two inputs are represented in parallel which means each input bit is available simultaneously. A serial-to-parallel multiplier processes the bits sequentially as opposed to multiplying every bit simultaneously. The LSB of one of the numbers, let's say B, is the beginning point, and it is multiplied by each bit of the other number, let's say A, starting with the LSB of A. After each multiplication operation, the multiplier keeps moving bits of B to the left. The result bits are simultaneously moved to the left to align with B's present location. The current bit of B is multiplied with all the bits of A at each step using an AND gate. As a result, the partial product is generated for that bit location. Finally, these partial products are added to produce a final result that is the product of two binary numbers. The final output bits are stored in accumulator unit. As B is processed, all of its components are added to the accumulation and B is shifted to the left as the process progresses. The sum of results in memory unit indicates the result of multiplying all the bits of B by all the bits of A. This multiplier has the benefit of reducing time delay since it produces the result in parallel. Power consumption will also be lower and speed of operation will improve due to low propagation delay. Figure 3 illustrates the model structure of SPM.

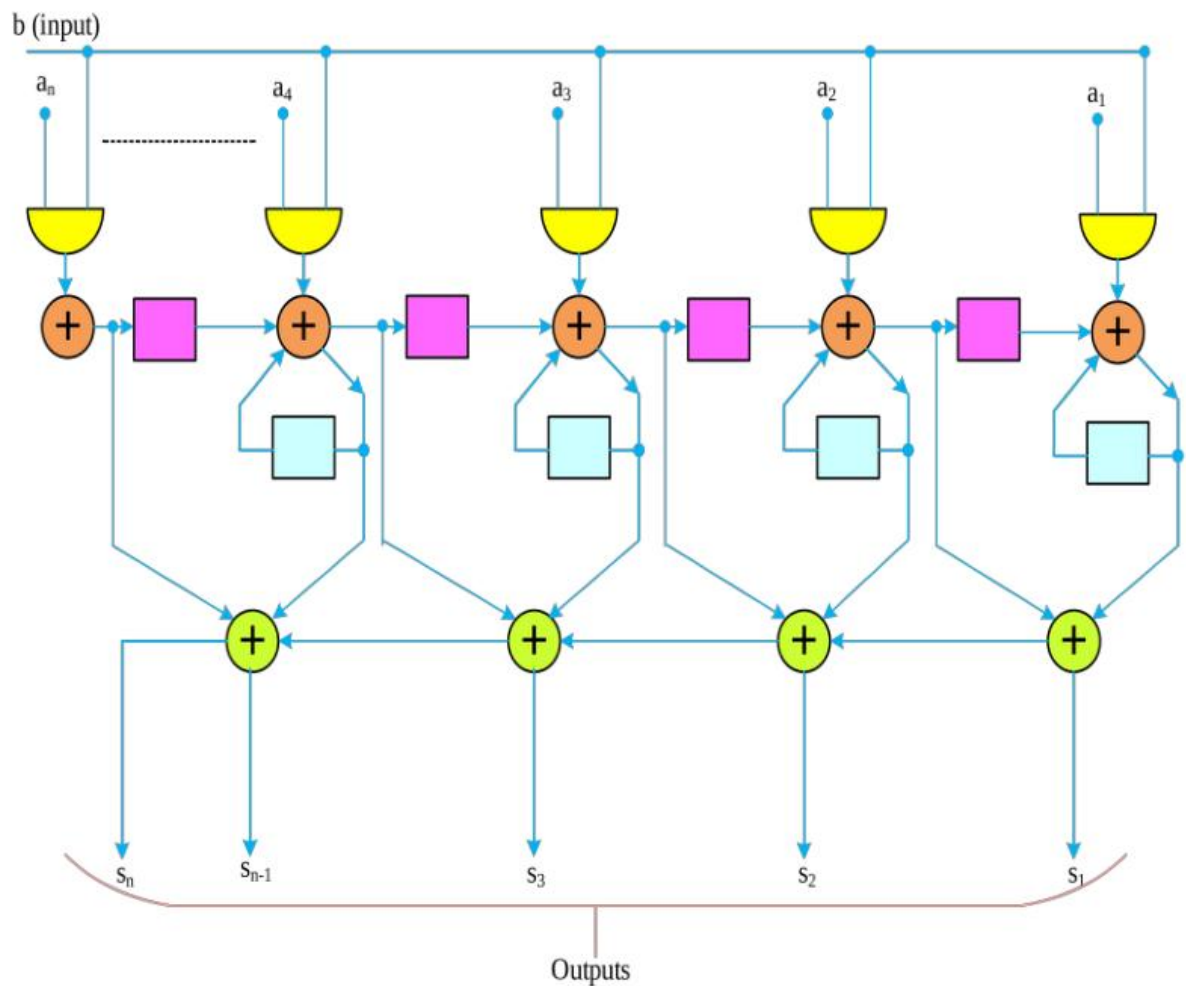


Figure 3. Architecture of Serial to Parallel Multiplier

The Montgomery multiplication is mainly used in cryptography technique to encrypt the data. Multiplication is a crucial part in ECC. The Montgomery multiplication is used in ECC to improve the speed of multiplication process. However, this technique has some limitations like it includes additional subtraction process. These extra steps in multiplication process may take longer time to complete the process. Due to this, the propagation delay will increase and speed performance will reduce. To overcome these drawbacks, SPM is introduced in Montgomery multiplication in this research. This technique will reduce time delay and increases the operating speed. So that power consumption of system obviously reduced. Figure 4 shows the diagram of Montgomery multiplication using SPM technique.

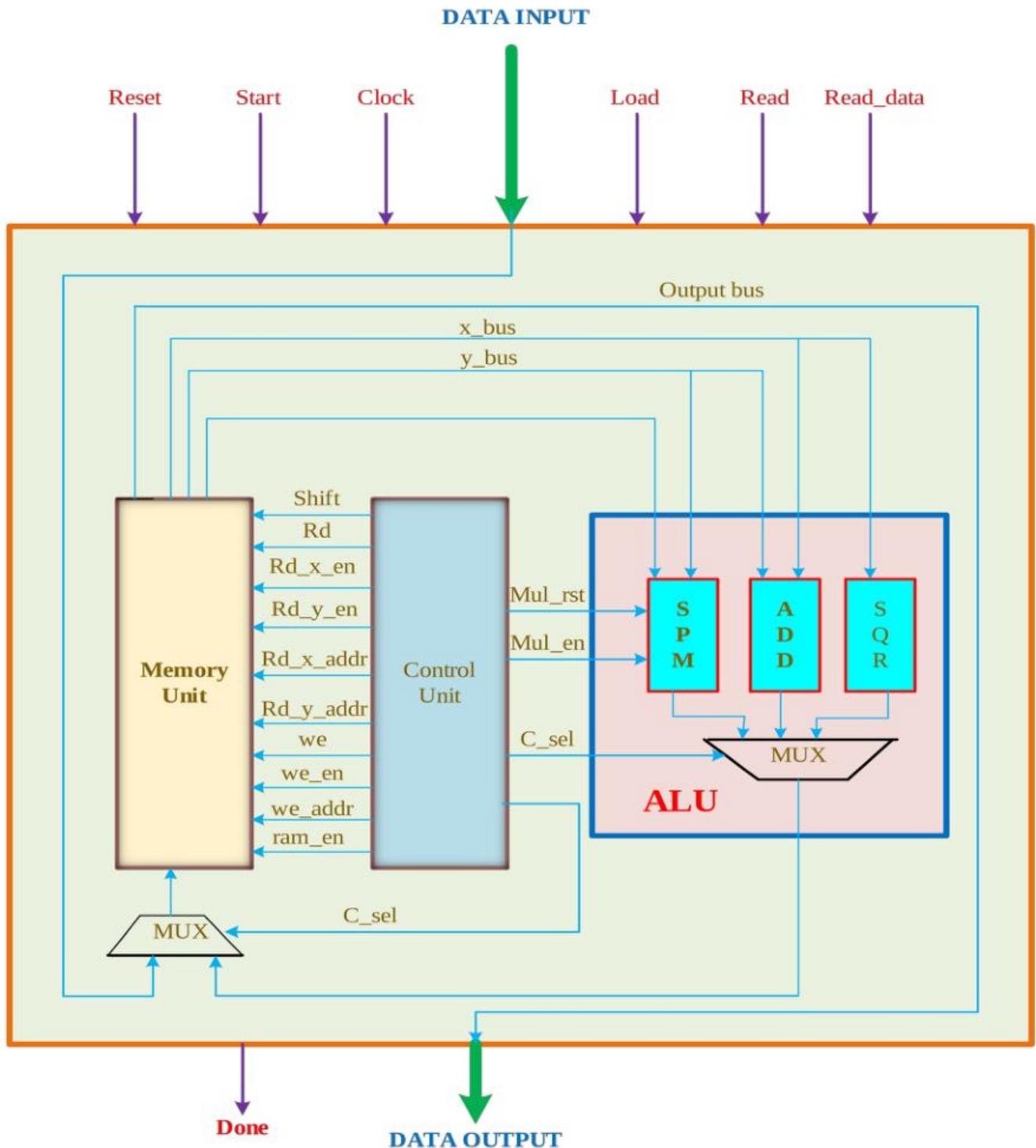


Figure 4. Architecture of Montgomery Multiplication with SPM for ECC

The goal of a design is to speed up the Montgomery multiplication used in ECC while using minimal resources by using proposed SPM in the multiplication process. In this work, the step of Montgomery multiplication method is replaced with serial to Parallel Multiplier strategy. This SPM is mainly

suggested to reduce latency and delay. This system includes memory unit, control unit, adder, multiplexer, and multiplier parts. The input data is stored in memory unit through multiplexer and control unit controls all signals used for process such as read, enable, and shift lines. The SPM is proposed here instead of traditional multiplier used in Montgomery multiplication process. Additionally, this system has reset, clock and load signals. The load line is applied before an input data is given to register. The result of SPM is stored in accumulator and produce it in parallelly. So, this proposed method provides more efficiency in ECC.

## RESULTS AND DISCUSSION

The proposed design was synthesized and simulated with Xilinx 14.7 ISE simulator tool using Verilog HDL. The proposed approach was used in Montgomery multiplication for ECC to improve the speed by reducing time delay. The main parameter for analysing speed and area of system is time delay and area. These parameters are evaluated in this chapter and obtained results were compared with existing models.

### Simulation of Serial to Parallel Multiplier

Initially, the 8-bit serial to parallel multiplier is synthesized and simulated to verify its functionality. Figure 5 illustrates simulation waveform of 8-bit SPM.

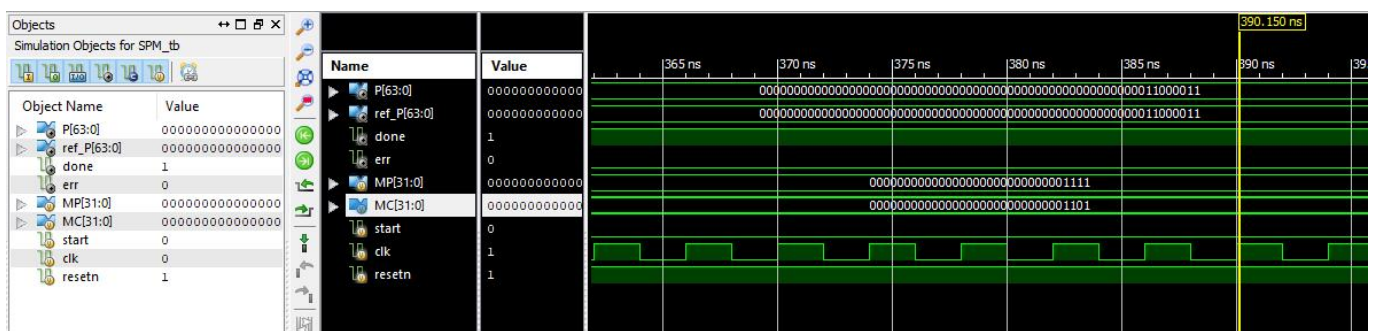


Figure 5. Simulation result of Serial to Parallel Multiplier

### Simulation result of Proposed SPM with Montgomery multiplication

In this study, the Serial to Parallel Multiplier approach is introduced in Montgomery multiplication instead of traditional multiplier. This Montgomery multiplication is mainly to used for cryptography to

encrypt information. To reduce delay and increase speed of the device the SPM is used for multiplication in ECC. The proposed design was synthesized and simulated using Xilinx tool and obtained results were indicated in this study. The two inputs x and y are 192 bits applied for multiplication. Additionally, clock, reset, and start signals were applied to do the process. The registers are used to store the inputs and results of multiplier. The resultant images are shown in following figures 6 -8.

## Comparison of Proposed method with RSA algorithm

The newly introduced SPM into Montgomery design was compared with existing strategy to analyse the performance of proposed design. The parameters such as LUT, FF, IOB, BUFG and delay were used to analyse the performance of the new design. This strategy is contrasted with the most recent one that was previously introduced. The proposed strategy was initially synthesized for 8-bit inputs and output waveform is displayed in Figure 6.

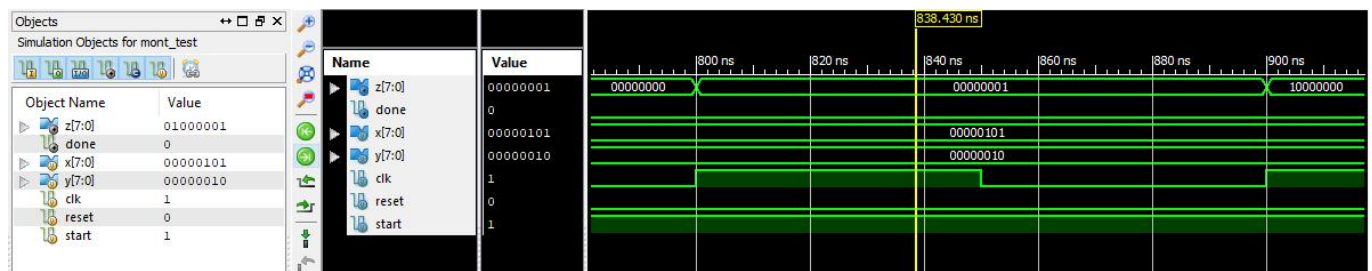


Figure 6. Simulation result of 8-bit proposed SPM Montgomery

In this work, the two 8-bit values are applied to the inputs x and y. The code was developed in Verilog HDL. The modulus input of this multiplication remains constant which is applied through code. The circuit multiplies these two 8-bit inputs and generates the remainder as result. The synthesis report was obtained and values are compared with most recent existing RSA Montgomery algorithm. Table 2 depicts the comparison results of proposed and existing method.

Table 2. Comparison of proposed with existing RSA Montgomery algorithm

| Method                                    | Size (bits) | LUT | FF | IOB | BUFG |
|-------------------------------------------|-------------|-----|----|-----|------|
| Montgomery using RSA algorithm, [25] 2023 | 8           | 64  | 24 | 35  | 1    |

|          |   |    |    |    |   |
|----------|---|----|----|----|---|
| Proposed | 8 | 51 | 33 | 28 | 1 |
|----------|---|----|----|----|---|

The proposed SPM Montgomery approach is evaluated and resultant values are compared with existing RSA algorithm which is recently introduced. In this analysis, the proposed method has great improvement than existing RSA algorithm. Initially, the design was synthesized and simulated with 8-bits. The value of LUT, FF, Input Output Block (IOB), and Clock Buffer (BUFG) of proposed design was listed as 51, 33, 28, and 1 respectively while it was indicated as 64, 24, 35, and 1 for RSA Montgomery algorithm. This result indicates that, the new design has improved performance and thus this technique is well suited for ECC for encrypting data.

## Simulation result of 192-bit proposed SPM Montgomery multiplication.

Figure 7 below depicts the RTL design of proposed SPM Montgomery Multiplication architecture. In hardware description languages (HDLs) like Verilog and VHDL, register-transfer-level extrapolation is used to build high-level models of a circuit from which lower-level views and, finally, actual routing, may be deduced. Figure 8 illustrates the top-level ports of proposed method.

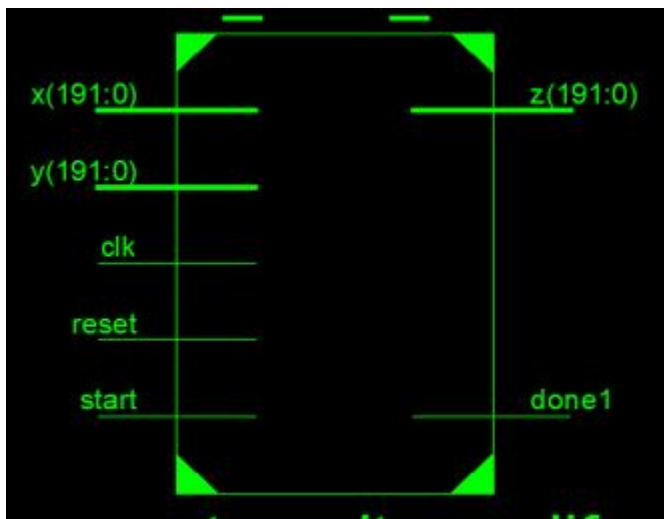


Figure 7. RTL schematic of Proposed method

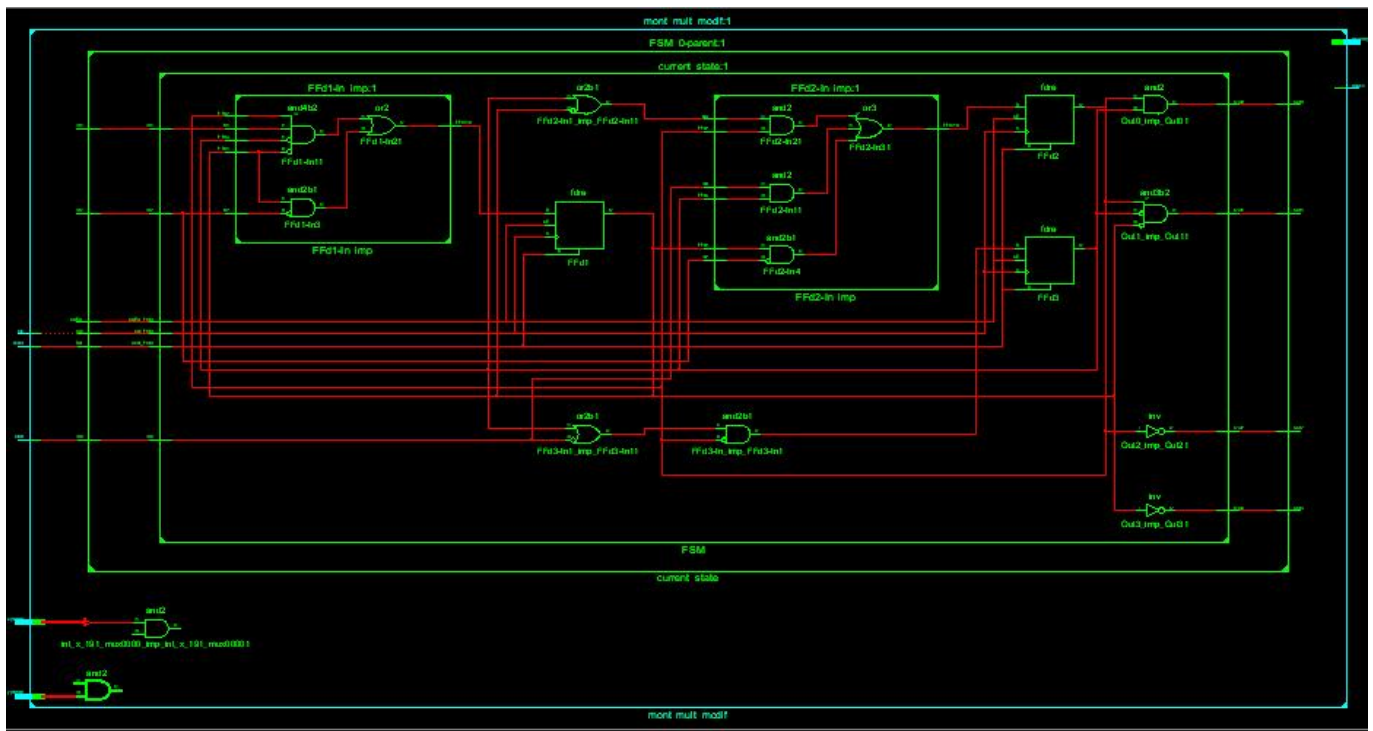


Figure 8. Top Level Ports of Proposed Technique

Figure 9 shows the simulation waveform of proposed SPM with Montgomery. This output waveform is generated using the code written in Verilog HDL. The two 192-bit inputs  $x$  and  $y$  are applied to the system and fixed modulus value is applied in code. The multiplication process was performed and remainder was generated as result( $z$ ). The reset, clock, and start signals were applied to perform the process.

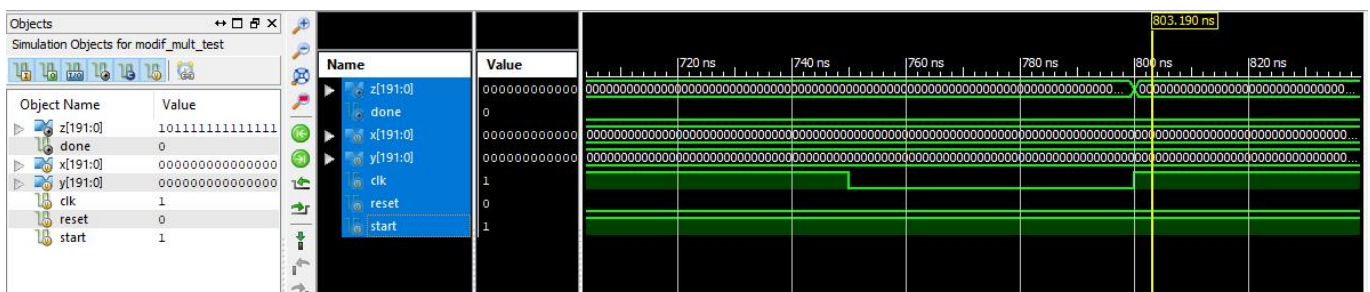


Figure 9. Simulation Result of Montgomery with SPM

## Analysis of Proposed design with various existing methods

The Proposed SPM Montgomery multiplication is evaluated and results were compared with several

existing techniques which are currently in use. This section covered the performance analysis of proposed design and the obtained results were compared with some existing methods. The main objective of this study is to reduce time delay and reduce the design area. In this analysis, the parameters such as LUT, time delay and frequency were computed to verify the performance of newly designed method. Time delay is the most crucial parameter for improving system performance. The power consumption of design will reduce due to less time delay. LUT is another parameter to decide the area of design and it may vary according to complexity of circuit design. The proposed design was initially synthesized using 8-bits and it was discussed in previous section of this paper and the results were compared with existing RSA Montgomery algorithm. It was clearly said that, the suggested strategy in this study has great performance when compared to RSA algorithm. To improve the efficiency of proposed design, further the proposed design was synthesized using 192- bits and the time delay and frequency was noted. The obtained results were compared with some methods which are currently in practice such as Karatsuba, Reversible, modular Montgomery, and one-dimensional semi systolic. These aforementioned methods were used in Montgomery multiplication for ECC. Table 3 illustrates the evaluation of existing and proposed design.

Table 3. Comparison Proposed design with several existing method

| Ref      | Method            | LUT   | Delay Time (ns) | Frequency (MHz) |
|----------|-------------------|-------|-----------------|-----------------|
| Proposed | SPM               | 5472  | 4.36            | 229.361         |
| [24]     | Karatsuba         | 6934  | 14              | 125             |
| [23]     | Reversible        | –     | 22.6            | 160             |
| [22]     | 1-D Semi-systolic | –     | 6.7             | 10              |
| [21]     | Modular           | 16241 | 19.3            | 135             |

The aforementioned table represents the comparison results. The proposed design has improved performance in all parameters. The LUT count of the design determines the area of circuit design. Some previous research study does not consider the LUT in their evaluation. However, this research evaluates LUT with few existing methods. From the analysis results, the newly developed Montgomery method has less LUT when compared to existing design such as Karatsuba and Modular Montgomery. The main parameter of this study is time delay and frequency determines the time delay. Therefore, the frequency and time delay are analysed and discussed in following section. Figure 10 displays the evaluation of time delay.

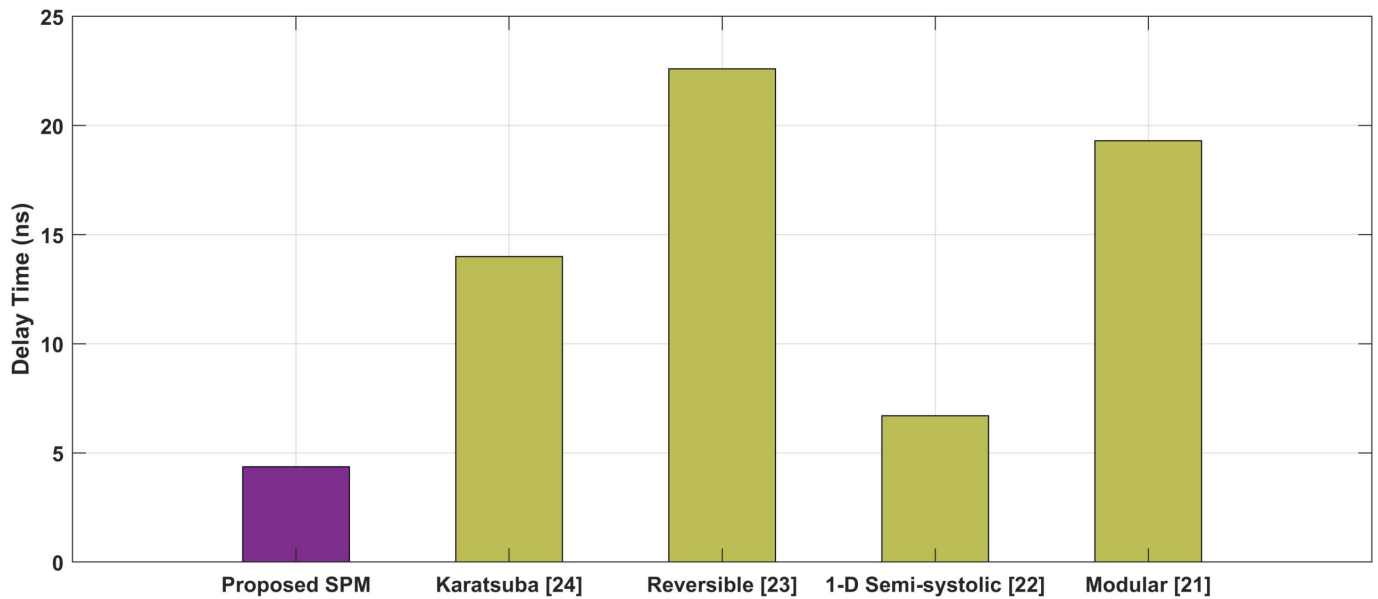


Figure 10. Analysis of Time delay

The aforementioned figure represents a time period comparison of existing and proposed design. The time delay is mentioned in nano seconds. The delay of Modular, 1-D semi systolic, Reversible, Karatsuba, and proposed Montgomery multiplication design are indicated as 19.3, 6.7, 22.6, 14, and 4.36 respectively. From the computation results, the newly developed design had a shorter delay than all other compared existing designs. This delay determines the power consumption and the analysis results clearly show that the proposed method will obviously reduce power since it produces least time delay. These delay values show that suggested strategy in this study has improved performance than other existing designs.

The processing time will be less when frequency increases. Therefore, the frequency is the second important parameter used to analyse the system performance. The frequency of developed design is evaluated and obtained values are compared with several design which are currently in use. The design with a higher frequency will have good performance. Figure 11 displays the frequency of proposed and existing designs.

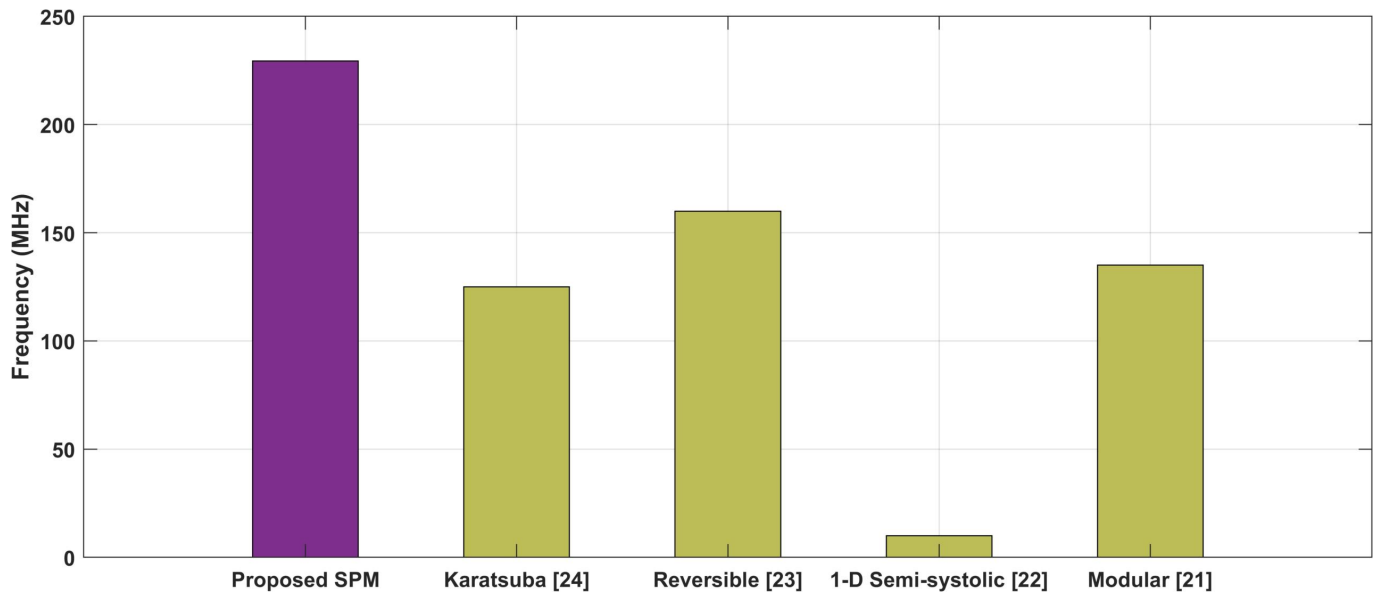


Figure 11. Analysis of Frequency

The above graph was plotted from the values given in Table 3. The frequency value for proposed, Karatsuba, Reversible, 1-D Semi-systolic, and Modular Montgomery multiplication are indicated as 229.361, 125, 160, 10, and 135 respectively. The above frequency values are expressed in MHz. From the analysis results, the proposed design has the highest frequency when compared to all other existing techniques. The design with highest frequency achieves great performance in time delay, and speed of the circuit.

From the above evaluation, the proposed design achieves less delay, more LUT and frequency. The analysed parameters achieve the better value in all parameters when compared to various existing designs. Therefore, the suggested design in this study has great performance and speed of the system will be high since it has the shortest delay. This proposed SPM Montgomery multiplication design is well suited for ECC to encrypt information with less time and high speed.

## CONCLUSION

In this article, the high speed and low area Montgomery multiplication using Serial to Parallel Multiplier for ECC was designed. The proposed design was synthesized and simulated using Xilinx ISE 14.7 simulator tool. The 8-bit SPM with Montgomery was synthesized and results were compared with RSA algorithm. In this comparison the SPM Montgomery has great performance when compared to RSA. Then the proposed design was implemented in 192-bits and obtained results were compared with various existing designs. The suggested processor's implementation findings revealed the

shortest data route at a maximum frequency of 229.361 MHz, which is the quickest recorded in the literature for completely LUT-based ECC processors. The processor can do a SPM Montgomery multiplication in 4.36 nanoseconds which is much less than several existing Montgomery multiplication methods. The main advantage of a proposed design is that it has a shorter delay and uses area efficiently which increases the operating speed in Montgomery multiplication used in ECC. Therefore, the proposed design is suitable for ECC to encrypt data with a shorter delay and a higher operating speed.

## References

- Fernandez-Carames, T.M. and Fraga-Lamas, P., 2020. Towards post-quantum blockchain: A review on blockchain cryptography resistant to quantum computing attacks. *IEEE access*, 8, pp.21091-21116.
- Rashed, H., Mohamed, E., Sistu, G., Kumar, V.R., Eising, C., El-Sallab, A. and Yogamani, S., 2021. Generalized object detection on fisheye cameras for autonomous driving: Dataset, representations and baseline. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (pp. 2272-2280).
- Sirisinahal, S., Murthy, M.R. and Ramu, S.C., 2021. Usage of Elliptic Curve Cryptography Technique for the Data Security and Privacy in the Wireless Sensor Network. *NVEO-NATURAL VOLATILES & ESSENTIAL OILS Journal/ NVEO*, pp.6208-6213.
- Rahman, M.S., Hossain, M.S., Rahat, E.H., Dipta, D.R., Faruque, H.M.R. and Fattah, F.K., 2019, February. Efficient hardware implementation of 256-bit ECC processor over prime field. In *2019 international conference on electrical, computer and communication engineering (ECCE)* (pp. 1-6). IEEE.
- Prabhu Kavin, B., Ganapathy, S., Kanimozhi, U. and Kannan, A., 2020. An enhanced security framework for secured data storage and communications in cloud using ECC, access control and LDSA. *Wireless Personal Communications*, 115, pp.1107-1135.
- Liu, S.G., Chen, W.Q. and Liu, J.L., 2021. An efficient double parameter elliptic curve digital signature algorithm for blockchain. *IEEE Access*, 9, pp.77058-77066.
- Rashid, M., Hazzazi, M.M., Khan, S.Z., Alharbi, A.R., Sajid, A. and Aljaedi, A., 2021. A novel low-area point multiplication architecture for elliptic-curve cryptography. *Electronics*, 10(21), p.2698.
- Longa, P., 2022. Efficient Algorithms for Large Prime Characteristic Fields and Their Application to

Bilinear Pairings and Supersingular Isogeny-Based Protocols. *IACR Cryptol. ePrint Arch.*, 2022, p.367.

Elkhatib, R., Azarderakhsh, R. and Mozaffari-Kermani, M., 2020, June. Highly optimized montgomery multiplier for SIKE primes on FPGA. In *2020 IEEE 27th Symposium on Computer Arithmetic (ARITH)* (pp. 64-71). IEEE.

Protzenko, J., Parno, B., Fromherz, A., Hawblitzel, C., Polubelova, M., Bhargavan, K., Beurdouche, B., Choi, J., Delignat-Lavaud, A., Fournet, C. and Kulatova, N., 2020, May. Evercrypt: A fast, verified, cross-platform cryptographic provider. In *2020 IEEE Symposium on Security and Privacy (SP)* (pp. 983-1002). IEEE.

Abebe, A.T., Shiferaw, Y.N. and Kumar, P.S., 2020. Efficient reconfigurable integrated cryptosystems for cybersecurity protection. *Advances in Cyber Security Analytics and Decision Systems*, pp.57-77.

Tian, J., Wu, B. and Wang, Z., 2021. High-speed FPGA implementation of SIKE based on an ultra-low-latency modular multiplier. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 68(9), pp.3719-3731.

Ellaithy, D.M., El-Moursy, M.A., Zaki, A. and Zekry, A., 2019. Dual-channel multiplier for piecewise-polynomial function evaluation for low-power 3-D graphics. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 27(4), pp.790-798.

Anderson, M., Chen, B., Chen, S., Deng, S., Fix, J., Gschwind, M., Kalaiah, A., Kim, C., Lee, J., Liang, J. and Liu, H., 2021. First-generation inference accelerator deployment at facebook. *arXiv preprint arXiv:2107.04140*.

Giao, J., Nazarenko, A.A., Luis-Ferreira, F., Gonçalves, D. and Sarraipa, J., 2022. A Framework for Service-Oriented Architecture (SOA)-Based IoT Application Development. *Processes*, 10(9), p.1782.

Li, J., Li, Z., Cao, S., Zhang, J. and Wang, W., 2019. Speed-oriented architecture for binary field point multiplication on elliptic curves. *IEEE Access*, 7, pp.32048-32060.

Mehrabi, M.A., Doche, C. and Jolfaei, A., 2020. Elliptic curve cryptography point multiplication core for hardware security module. *IEEE Transactions on Computers*, 69(11), pp.1707-1718.

Rashid, M., Imran, M. and Sajid, A., 2020. An efficient elliptic-curve point multiplication architecture for high-speed cryptographic applications. *Electronics*, 9(12), p.2126.

Abd-Elkader, A.A., Rashdan, M., Hasaneen, E.S.A. and Hamed, H.F., 2020. FPGA-based optimized design of montgomery modular multiplier. *IEEE Transactions on Circuits and Systems II: express*

*briefs*, 68(6), pp.2137-2141.

Zhao, X., Li, B., Zhang, L., Wang, Y., Zhang, Y. and Chen, R., 2021. FPGA implementation of high-efficiency ECC point multiplication circuit. *Electronics*, 10(11), p.1252.

Ahmadpour, Z. and Jaberipur, G., 2021. Up to \$8 k \$8 k-bit Modular Montgomery Multiplication in Residue Number Systems With Fast 16-bit Residue Channels. *IEEE Transactions on Computers*, 71(6), pp.1399-1410.

Ibrahim, A., Tariq, U., Ahanger, T.A. and Gebali, F., 2022. Low-Complexity One-Dimensional Parallel Semi-Systolic Structure for Field Montgomery Multiplication Algorithm Perfect for Small IoT Edge Nodes. *Mathematics*, 11(1), p.111.

Charishmmaa, G., Reddy, S.K. and Guravaiah, V., 2022. Design Of Power And Area Efficient ECC Processor Using Reversible Technique.

Awaludin, A.M., Park, J., Wardhani, R.W. and Kim, H., 2022. A high-performance ecc processor over curve448 based on a novel variant of the karatsuba formula for asymmetric digit multiplier. *IEEE Access*, 10, pp.67470-67481.

Jain, A. and Bansod, P.P., 2023. FPGA Implementation of Enhanced Montgomery Modular for Fast Multiplication. In *ITM Web of Conferences* (Vol. 56, p. 01001). EDP Sciences.