Implementation of Multifunction Residue Architectures on FPGA for Cryptography Applications

Dr.A.Pradeep Kumar[1], V. Srinivas[2]

[1]Assistant Professor, Malla Reddy Engineering College (Autonomous) Main Campus

[2] Assistant Professor, Malla Reddy Engineering College (Autonomous) Main Campus

Abstract: Data confidentiality and encryption are essential features of modern computing. Due to its high limited size and therefore its reliability equal to that of other traditional public-key algorithms, elliptical curve cryptography (ECC) is the ultimate cryptographic method. Furthermore, the hardware acceleration of cryptographic algorithms is necessary to satisfy the raising speed demand for contemporary implementations. The core component of the residue number method is the forward converters, modulo decimal units and the reverse converters. The reverse converter is based using standard and compact adders in the current network of residue numbers. It displays substantial power usage and low speed. In-depth, it suggested a residue Montgomery algorithm for multiplying data paths among converters and between the two residue representations by evaluating the input/output conversion to/from residue representation. The design is acquired thus, we use the dual field modulator for the GF(p) and GF(2)n Montgomery multiplication, input/output conversions, mixed-radius conversions (MRC) for inputs and polynomials and exponentiation and inversion in the same hardware.

Keywords: Research in finite fields, computer arithmetic, Montgomery multiplication, parallel mechanism in arithmetic and logic structure.

I. INTRODUCTION

Throughout this era, wireless and digital interactions have rising astronomically. Hundreds of purchases exist on both the worldwide network daily. Almost all of these activities include important, sensitive details, verified transactions and authenticated users. Such specifications include a comprehensive protection system.With the rapid development of secure network communications, cryptography criteria for authentication have recently raised exponentially. Modular multiplication is a basic practice in many common architectures including RSA [1] and ECC [2, 3] for public-key encryption (PKC). Since the division process in modular reduction takes time, Montgomery [4] suggested a new methodology to prevent division. ECC is a very popular and powerful public-key encryption method for cryptographic purposes and is now very prominent in both primary and binary fields due to the smaller field size. The focus of this research will become an elliptical curve over a binary field because it is quite effective for hardware deployment due to the use of modulo 2 arithmetic.

An elliptical curve over a finite field gives a group structure used to enforce the cryptographic scheme. The activities of the company are PD and PA. Such two community operations have been merged into a lightweight hardware architecture and named PDPA. For elliptical group curve procedures, two well-known coordinate systems are often used: Affine coordinate structures and projective coordinate structures. Furthermore, the most sophisticated modular multiplication methods in Montgomery were designed for the set accuracy of operands which cannot be used for variable accuracy. Different strategies for improving the performance of point multiplication are implemented using either FPGA or ASIC implementation, including algorithm optimization and enhanced arithmetic field architectures. However, most point multiplication systems have been introduced with independent group operations, which could improve the frequency of group activities and thus decrease the

multiplication pace. Although several high-speed point multiplication strategies were described in the literature for an ECC processor, most are field-efficient. Our suggested design has a speed-area trade-off which is ideal for quicker, new, cryptography implementations. Practices including such multi-module and multi-function models have been proposed to eliminate redundancies in the hardware and therefore to mitigate power dissipation through multi-threshold voltage and multi-source stress architectures. These optimization techniques are designed for the exploration of space at the algorithm level and are specific to common numerical modules. To simplify the design level of such modulo arithmetic processes like the multiplication of units, techniques that investigate particular theoretical characteristics of the different modules of types $2n$ and $2n \pm 1$ have received comprehensive attention, amongst many others.

The VLSI design recommended in this article for the usage of RNS is the macro selection criterion for the chip memory use, which contributes to the complex implementation of main techniques. The literature survey indicates that various working conditions were suggested for the multiplication of cryptography by companies focused on polynomials to address the shortcomings of RNS schemes. The algorithm of Montgomery is one of the master areas of dual-field applications. Under key RSA word lengths, the Montgomery architecture fits well with the processing of word-size details, considering that key RSA sizes (512, 1024, 2048, etc.) often have a multiple word scale. Furthermore, the main sizes in error correction techniques are not integer multiples, therefore it indicates because, if such designs have been used in the error correction application techniques, more clock cycles would be needed to enforce them, which would result in higher power usage. This issue may be solved by implying that the system is designed at a bit level. This method is designed and applied with a view to the environment and power consumption; the next portion of the process summarizes so much about the framework suggested.

The main contributions and organization of this paper are summarized as follows: In section 2 we describe background details of Montgomery operations. Section 3 discusses the proposed work. Section 4 deliberates Results and Discussions. Finally, in section 5, we concluded the paper.

II. BACKGROUND WORKS

Encryption is performed through a public key, so then the matching private key will decrypt the encrypted document. The reliability of these algorithms depends on the hardness of the public key to extract the private key. While gradual and highly technical, asymmetric key encryption provides tremendous benefits. The main benefit seems to be that the simple primitives used will be based on well-known issues like integer factoring and discrete problems with the logarithm. Such difficulties were extensively studied and, despite years of research, their consistency was not contradicted. This may be distinct from symmetric key encryption, in which the power of the code relies on combinatory strategies. The protection of such algorithms has not been demonstrated and therefore does not rely on well-researched literary problems.

Within real-world environments, crypto-system installations must be effective, scalable and reusable. Implementations such as smart cards and mobile phones need deployment where resource consumption and power consumption are important. Such architectures should always be lightweight and low-powered. A secondary criterion is computing capacity. The degree of system reconfigurability may also be held to a minimum. This seems to be since these machines are short-lived and are typically only installed once. High-efficiency devices such as network servers, storage networks, etc. on the other side of the continuum need high-speed deployment of ECC. The crypto algorithm will never be the output bottleneck of the program. These architectures do need to be rather scalable. Running conditions can be reconfigurable, such as algorithm variables, etc. The software can be quickly reconfigured; furthermore, program implementations do not often scale the application's output. These systems need specialized equipment for speed calculations. The required clock cycles, operating frequency, and region are critical design parameters while utilizing these hardware accelerators. The clock cycles will be short and the frequency high, so that the average device latency is smaller. The region is essential since smaller areas imply that parallelism on the same equipment may be introduced, thus the device's performance.

The Correctness of RSA). Let *X*, *Z*, *Y*, *l*, *m* be plaintext, cipher text, encryption exponent, decryption exponent and modulus, respectively. Then

$$(X^{l})^{m} \equiv X \left(mod\, Y\right) \tag{1}$$

Proof. Notice first that

$$Z^{m} = (X^{l})^{m} X \left(mod\, Y\right) \tag{2}$$

$$Z \equiv X^{l} \left(mod\, Y\right)$$

$$\equiv X^{1+k\phi(Y)} \left(modY\right) (lm \equiv 1\left(mod\phi\left(Y\right)\right) \tag{3}$$

$$\equiv X.X^{k\phi(Y)} \left(modY\right) a^{\phi(y)} \equiv 1\left(modY\right)) \equiv X \tag{4}$$

The result thus follows.

Remark 1. In order to complete, this process of decryption, the approved users having the *m*, *r* and *s*, with the data available from $X \equiv Z^{m} \left(mod\, Y\right)$, makes the computation to operate at speed by taking into account two axioms

$$X_{r} \equiv Z^{d} \equiv Z^{m \bmod r-1} \left(mod\, r\right) \tag{5}$$

$$X_{s} \equiv Z^{d} \equiv Z^{m \bmod s-1} \left(mod\, s\right) \tag{6}$$

$$X \equiv X_{r}.s.s-1 mod r + M_{s}.r.r-1 mod\, s \left(mod\, Y\right) \tag{7}$$

Recently the residue number system $\{2^{y}\text{-}1, 2^{y}, 2^{y}\text{+}1, 2^{2ys+1}\text{-}1\}$ supposed to introduce both high dynamic terms with the logic of parallelism. The block diagram of RNS system based on these moduli set is shown in Figure1.

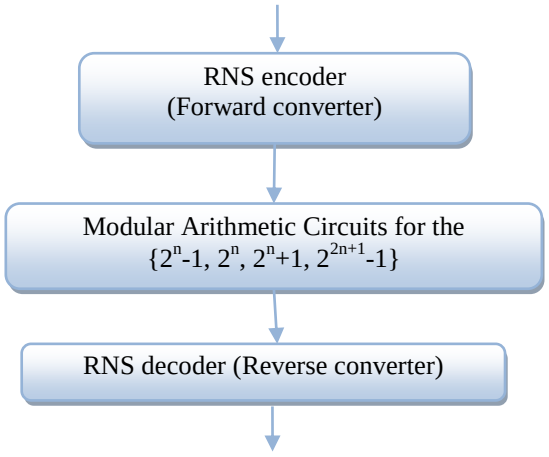

Figure 1: Block Diagram of residue number system

The first phase in constructing a converter is the collection of the module package. The collection of modules will play an important role in the dynamic range, pace, and hardware creation of the RNS. The second phase for constructing a reverse converter is to replace the variables of the module provided by modern CRT conversion algorithms.

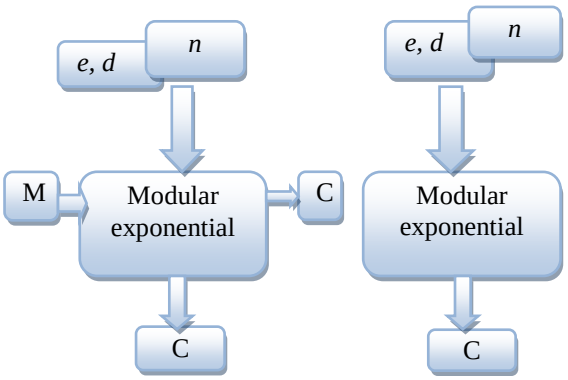

Figure 2 : The RSA encryption/decryption structure

The corresponding calculations are condensed in the third stage by utilizing mathematical characteristics and proposals. Eventually, adder elements including CSA-EAC, CPA-EAC, and Revised adders are created for final equations. Figure 2 shows the RSA encryption/decryption structure in hardware implementation. The essential of the RSA implementation is how better the modular arithmetic operations that can incorporates the modular addition, subtraction, multiplication and exponential.

III. METHODS AND METHODOLOGY

Dual-Field Modular Reduction: A final modular reduction by each modified RNS/PRNS modulus is required, for each multiplication outcome, within each MAC unit.

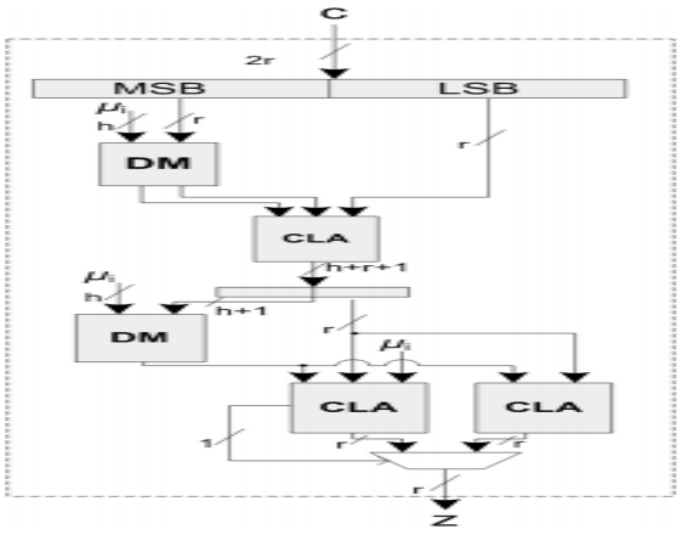

Figure 3: Connections of Dual-field modular and CLA units

Let us denote the 2*p*-bit product where its primary requirement is to modulo reduction p_i. By taking into consideration r_i and treating the term $2p-\mu_i$, in order to denote *h*-bit μ_i 2^p, the major calculation can be done which can be further be simplified as

$$z_{r_i} = \sum_{i=0}^{p-1} z_i 2^i + 2^p \sum_{i=0}^{p-1} z_{r+i} 2^i_{r_i} = G + 2^p H_{r_i} \quad (8)$$

$$= \sum_{i=0}^{r-1} m_i 2^i + \mu_i \sum_{i=0}^{h} m_{r+i} 2^i{}_{r_i} \quad (9)$$

A certain mathematical calculation extended for differential equations where a compact dual-field reduction (DMR), as seen in the figure, can be mechanized and adders, multipliers depends on modular operation are used. Nonetheless, CSKA, which would be an effective adder in terms of energy use and region usage, was implemented by CLA. The CSKA's vital latency is much lower than that of the RCA, whereas its region and electricity consumption are comparable to RCA. However, the CSKA power delay product (PDP) is smaller than the CSLA device.

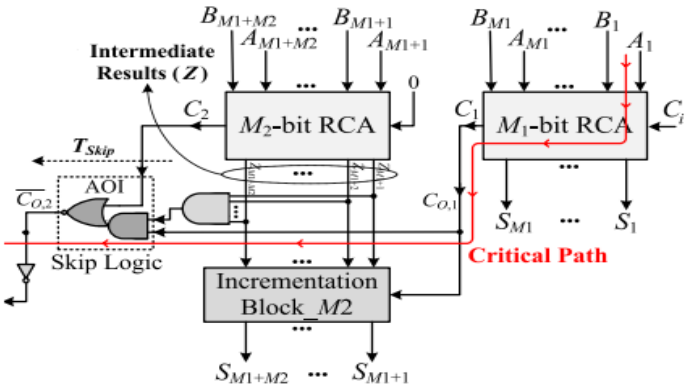

Figure 4: Modified CI-CSKA structures

In Figure 4, the modified CI-CSKA structure which can be substituted in Figure 3 by a CLA block allows us to use easier carrying skip logic. The AOI/OAI compound gates are quicker and more efficient. Remember that it is augmented in this form when the carry propagates through the skip logic. Furthermore, the completion of the carry is obtained at the output of the skip logic of even phases. The structure has a significantly lower propagation time with a slightly smaller area compared to existing areas on its own. Likewise, while the AOI gate's electric energy consumption is slightly smaller than that of the multiplexer, the proposed CI-CSKA's energy consumption is slightly higher than traditional power consumption.

IV. RESULTS AND DISCUSSION

The Montgomery architecture and modular RSA encryption was written in VHDL/Verilog and synthesized in the FF1153 speeds of XILINX ISE 14.4i (Kintex7) and XILINX ISE 12.4 products. The findings of the synthesis of the original Montgomery specification for various operand size bits are shown in Table 1-3. In the synthesis report region and frequency (minimum period) are produced. The output is measured as bit length multiplied by frequency and divided by clock cycle number.

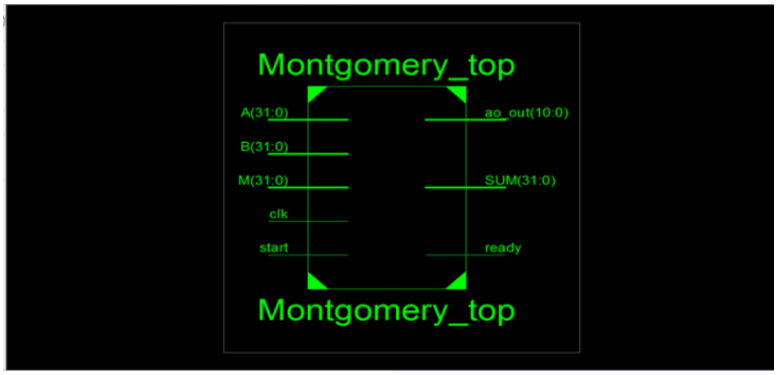

Figure 5: RTL schematic of proposed design

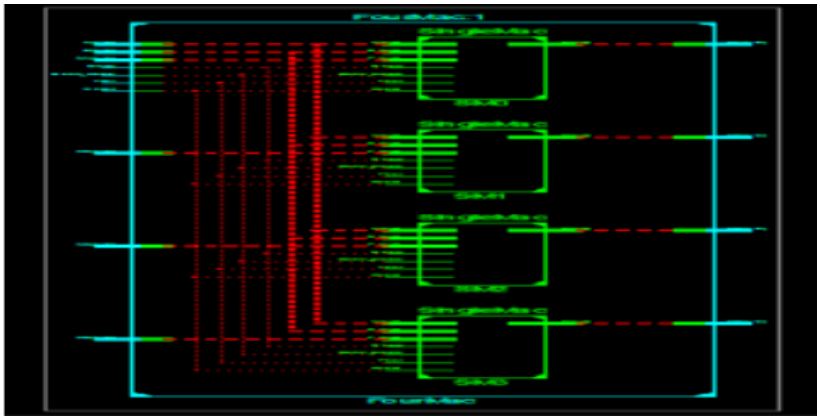

Figure 6: Technology schematic of proposed design

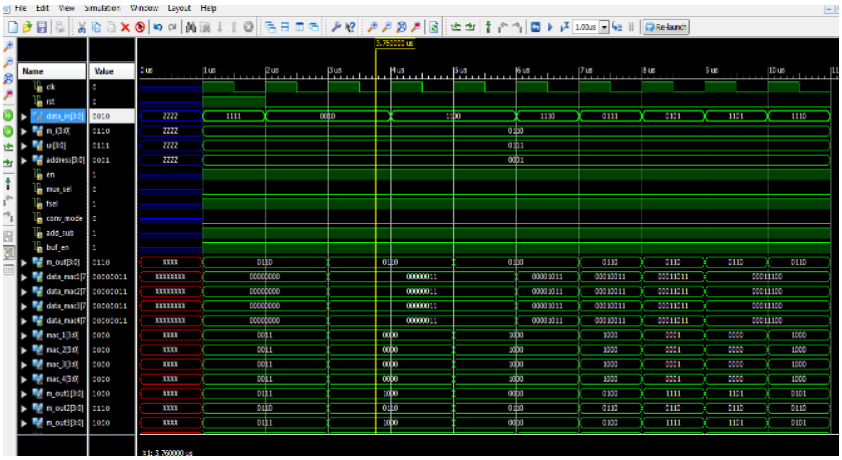

Figure 7: Simulation results of the proposed design

Table 1: Overview of logic blocks usage CSK adder for 4-bit length

Device Utilization Summary (estimated values) [-]			
Logic Utilization	Used	Available	Utilization
Number of Slices	5	960	0%
Number of 4 input LUTs	10	1920	0%
Number of bonded IOBs	14	66	21%

Delay in the path for CSK adder for 4-bit length is 9.23ns

Table 2: Overview of logic blocks usage CSK adder for 8-bit length

Device Utilization Summary (estimated values) [-]			
Logic Utilization	Used	Available	Utilization
Number of Slices	9	960	0%
Number of 4 input LUTs	15	1920	0%
Number of bonded IOBs	26	66	39%

Delay in the path for CSK adder for 8-bit length is: 9.23ns: 12.45ns

Table 3: Overview of logic blocks usage CSK adder for 16-bit length

Device Utilization Summary (estimated values)			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	14	960	1%
Number of 4 input LUTs	25	1920	1%
Number of bonded IOBs	50	66	75%

Delay in the path for CSK adder for 16-bit length is: 15.23ns

Table 4: Overview of logic blocks usage CSK adder for 32-bit length s

Device Utilization Summary (estimated values)			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	26	960	2%
Number of 4 input LUTs	45	1920	2%
Number of bonded IOBs	98	66	148%

Delay in the path for CSK adder for 4-bit length is: 31.23 ns

Table 5: Device Utilization Summary of 4, 8, 16, 32-Bit Carry's Skip Adder

Logic Utilization	**4 – Bit CSkA**	**8 - Bit CSkA**	**16 – Bit CSkA**	**32– Bit CSkA**
Number of Slices	5	9	14	26
Number of 4 input LUTs	10	15	25	45
Number of Bonded IOBs	14	26	50	98
Delay	8.7ns	11.3ns	14.1ns	29.8ns

V. CONCLUSION

This paper presents the design of an efficient RSA cryptosystem that uses the Montgomery algorithm for modular multiplication. With the help of dual field residue arithmetic unit, the Montgomery multiplication is done for all cases of word lengths are achieved. However, the associated DRAMM unit makes possible operations of multiplication in GF(p) and GF(2n), with the residual model. For improving the overall performance of the architecture of Montgomery and adder, the response time and critical path are decreased. The advantage of the proposed CI-CSKA achieved better results as related to other techniques.

REFERENCES

[1] R. L. Rivest, A. Shamir and L. M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 21(2):120-126, 1978.

[2] N. Koblitz. Elliptic curve cryptosystem. Math. Comp., 48:203-209, 1987.

[3] V. Miller. Uses of elliptic curves in cryptography. In H. C.Williams, editor, Advances in Cryptology: Proceedings of CRYPTO'85, number 218 in LNCS, pages 417-426.Springer-Verlag, 1985.

[4] P. Montgomery. Modular multiplication without trial division. Mathematics of Computation, 44:519-521, 1985.

[5] A. Tenca and Ç. K. Koç. A scalable architecture for modular multiplication based on Montgomery's algorithm. IEEE Transactions on Computers, 52(9):1215-1221, September 2003.

[6] J.-C. Bajard, L. Imbert, and G. A. Jullien, “Parallel Montgomery multiplication in using Trinomial Residue Arithmetic,” in IEEE Symp. Computer Arithmetic, 2005, vol. 0, pp. 164–171.

[7] N. Guillermin, “A high speed coprocessor for elliptic curve scalar multiplications over,” in Cryptographic Hardware and Embedded Systems, CHES 2010, 2010, pp. 48–64, Lecture Notes in Computer Science 6225.

[8] N. Mentens, K. Sakiyama, B. Preneel, and I. Verbauwhedes. Efficient Pipelining for Modular Multiplication Architectures in Prime Fields. Proceedings of the 2007 Great Lakes Symposium on VLSI (GLSVLSI 2007), 2007.