An Efficient Architecture for Secured Communication through Montgomery Modular Multiplication

V.Kavitha,
Professor, Department of Electronics and Communication Engineering,
M.Kumarasamy College of Engineering,
Karur, Tamilnadu, India.
emiroece@gmail.com

S.Selvi,
Assistant Professor, Department of Electronics and Communication Engineering,
M.Kumarasamy College of Engineering,
Karur, Tamilnadu, India.
selvi0412@gmail.com

***Abstract*—Security is the major concern in today's communication processes. Modular multiplication is one among those methods to provide secured communication from early time with longer delay and complexity. Montgomery modular multiplication (MMM) is the one which provides high security with reduced complexity. This paper proposes an architecture to enhance the existing Montgomery Modular Multiplication (MMM) by replacing Carry Save Addition (CSA) part with the Carry Skip Adder (CSKA) for reducing delay and with the improved Carry Select Adder (CSLA) for improving power and area. Comparisons are made with the results obtained to select the Montgomery Modular Multiplication depending on the requirement.**

Keywords—Communication; Montgomery modular multiplication; Carry Save Adders; Carry Skip Adders; Carry Select Adders; Delay.

I. INTRODUCTION

Many communication algorithms such as modulation, demodulation, phase shift keying etc requires multiplication operation in their processing. Normal multiplication operation is easier and can be computed in an easier manner. The secured communication can be achieved through cryptographic processes. Many of the cryptographic processes make use of the modular multiplication processes. Modular multiplication increases its complexity when the number of bits to be operated is very high.

In the modular conversion cryptographic algorithm, higher values of inputs are converted into lower values by taking modulo of their values with respect to a residue. For example, for a multiplication of two 8 bit numbers which would take much more time to produce the 16 bit product. When converted to a modulo value, the multiplication time is reduced [4].

To enhance the performance of this modular multiplication, a method was proposed by Montgomery [4] in 1985 which increases the speed of the modular multiplication. This Montgomery multiplication gained its interest in reducing the area occupied in its hardware architecture and to enhance its speed. Even though the main aim of Montgomery Modular Multiplication is to reduce the delay involved, it still involves with the time consuming carry Propagation in the Carry Save addition part.

Several architectures have been proposed for Montgomery modular multiplication from its invention. The basic Montgomery Multiplication uses Carry Save addition[7] and in further papers[11 -14], the Carry Save addition has been splitted into Semi Carry Save Addition and Full Carry Save addition.[5]-[8],[23]. The use of carry save addition reduces the area Delay product to a great extent than the normal modular multiplication [6].

The use of Carry Save adders other than the Ripple Carry Adder and Carry Look Ahead Adder reduces the latency and area of the circuit to a great extent. Taek Won Kwon[8] and Kim proposed a method where two CSA are used and the final product is obtained from a Carry Propagate adder along with a Shift register.

In [8], the architecture for MM multiplication uses the existing two Carry Save Adders to produce the final product also whereas it avoids the use of Carry Propagation Adder and the shift register. Hence fast summation occurs and the latency is reduced. Also the area has been reduced as the existing CSA is reused.

In [10], the use of MM multiplication in RSA cryptosystems has been discussed. In this paper, the intermediate superfluous operations have been bypassed. The initial Modular Multiplications had more number of iterations which consumed more time and this paper has reduced the number of iterations by using MM52 and MM42 algorithms. The energy consumption has also been reduced by using barrel shift addition technique and gated clock technique.

II. MONTGOMERY MODULAR MULTIPLICATION USING CARRY SAVE ADDITION

Enhancement of Montgomery Modular Multiplication through the use of other types of adders in place of the Carry Save Adder is focused as follows.

Instead of using two level FCS in MM multiplication, here a single level SCS based system. Also the time spent for conversion from normal format to modular format is taken care.

The first one is the MM1 method discussed in [5]. Here Carry Save adders along with a carry propagate adder which is of 32 bit length and this would take 32 clock cycles to produce the multiplication product is used. Area is the main issue to be taken care here. The following Fig. 1. depicts the MM1 algorithm with 2 level CSA.

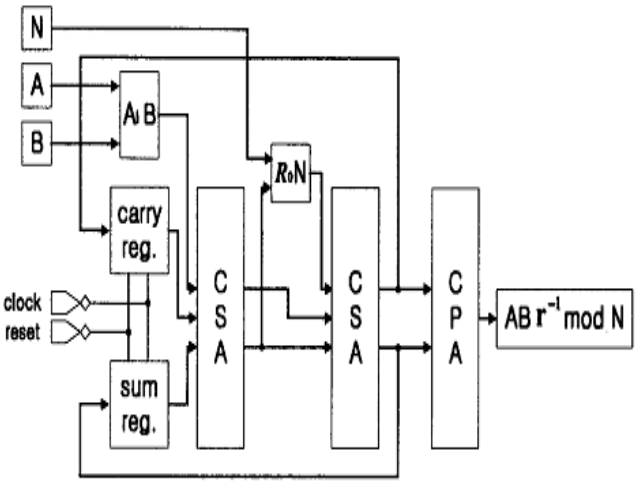

Fig. 1. MM1 with two level CSA along with CPA

In [6] & [7], a new modification which uses only two carry save adders alone and the use of carry propagation adder has been avoided by dividing it into two additions namely SC(K+2) and SS(K+2). Again it suffered the problem with the carry propagate mechanism. The architecture is illustrated in the following Fig. 2.

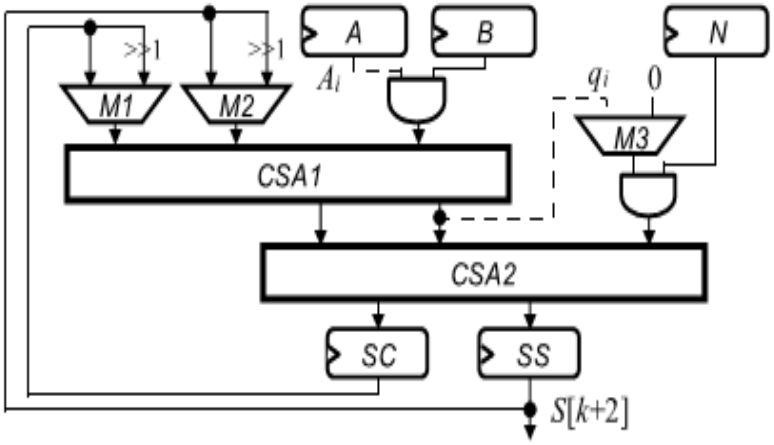

Fig. 2. MM2 with two level CSA

Further a new method was proposed with three level Carry Save Adder in [10]. Here A input is divided into two parts namely AS and AC and stored in two shift registers and then a Full adder is used along with a Barrel Register Full Adder (BRFA) whose concept is available in [10]. The architecture for Montgomery Modular Multiplication along with Barrel Register Full Adder is as shown in Fig. 3.

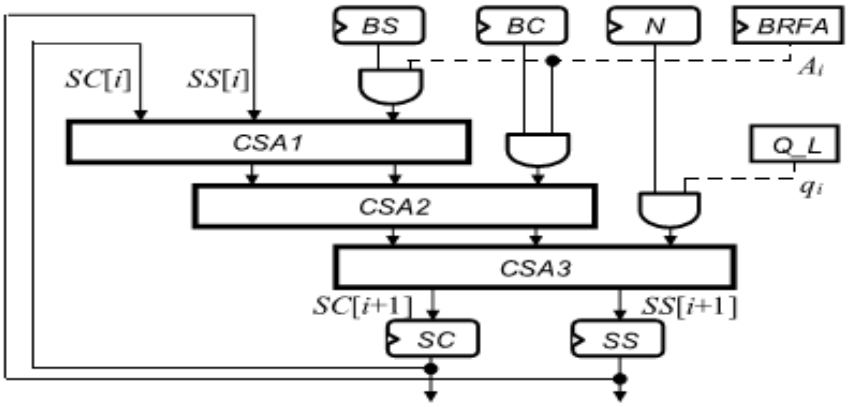

Fig. 3. FCS MM1 Multiplier with 3 level CSA and BRFA

In the above FCS MM 1 multiplier in order to reduce the latency, the inputs A and B and the output sum are represented in two carry save representations such as [AS,AC],[BS,BC] and [SS,SC]. This MM1 consists of a three level and a two level carry save architecture. The next one proposed was FCS MM2 architecture. Here at the starting of each modular multiplication, BS and BC are added with N and stored at DS and DC. Hence here it is a two level multiplier. But in order to store DS and DC it needs extra register. Hence the critical path is reduced by one carry save level at the cost of extra hardware.

Hence on comparing the above architectures, it is observed that SCS based MM multiplier can perform with a compensation of extra clock cycles for format conversion with an advantage of low hardware overhead. However with an excess in hardware overhead the delay of clock cycles for modular format conversion can be avoided as in FCS modular multiplier.

This paper aims to reduce the extra clock cycles required by using the SCS modular multiplier with the available low hardware overhead. Here the number of clock cycles can be reduced by using a comparison circuit called Zero-D circuit which checks whether the SC is zero or not by using a NOR gate logic. In this architecture, B+N addition is performed till SC=0 and also at the same time format conversion is also done. The delay is also reduced with the selection of input 0, N, B and D through the multiplexer M3 where A and qI can be pre computed in the (i-1)th iteration itself.

The following Fig. 4. describes the improved SCS based modular multiplier.

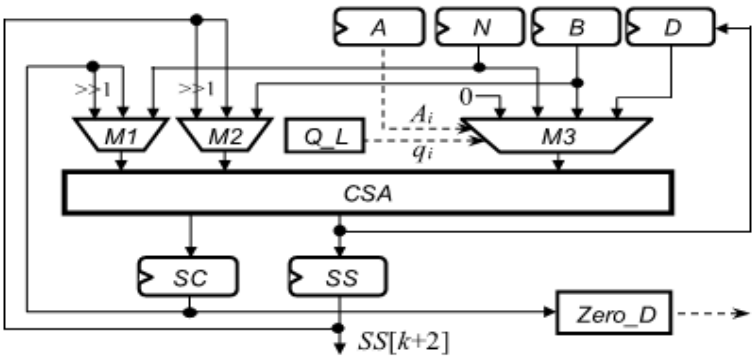

Fig. 4. Modified SCS multiplier.

The hardware overhead can be further enhanced by replacing the carry save adder in the above circuit by the area delay power efficient carry select adder.

III. Carry Select Adder

The working of normal carry skip adder is as follows:

In order to add a stream of two numbers, ripple carry adder (RCA), where the carry is rippled through the LSB to MSB is suggested. Even though this has a simple hardware architecture, the main drawback of this circuit is its carry propagation delay.

An advanced adder called Carry Select adder, where the time before the (n-1)th carry enters the nth bit adder, the adders themselves add and store the sum for both previous bit carry =0 and 1. From the two calculated sums, correct sum is chosen after the previous carry bit is received. This is done with a overhead in hardware.

The hardware overhead is further reduced by using a Binary to Excess 1 converter instead of using a separate adder with carry=1. This further reduces the area of the carry select adder and its power is also efficiently reduced.

This CSLA can be further enhanced by splitting the circuit. The Carry Select Adder has two units: Generation Unit and Selection Unit. The Generation Unit and the Selection Unit has been separately designed for Sum and Carry as Sum Generation (SG) Unit, Sum Selection (SS) Unit, Carry Generation (CG) Unit, Carry Selection (CS) Unit. For the purpose of simplifying the circuit further the SG and CG units are further divided into Half SG and Half CG and Combined through a Full Generation and Full Selection Unit. Its diagrammatic representation is shown as in the following fig. 5. The actual operation which has been done in this circuit is the redundant logical operations have been avoided and also the data dependency has been reduced.

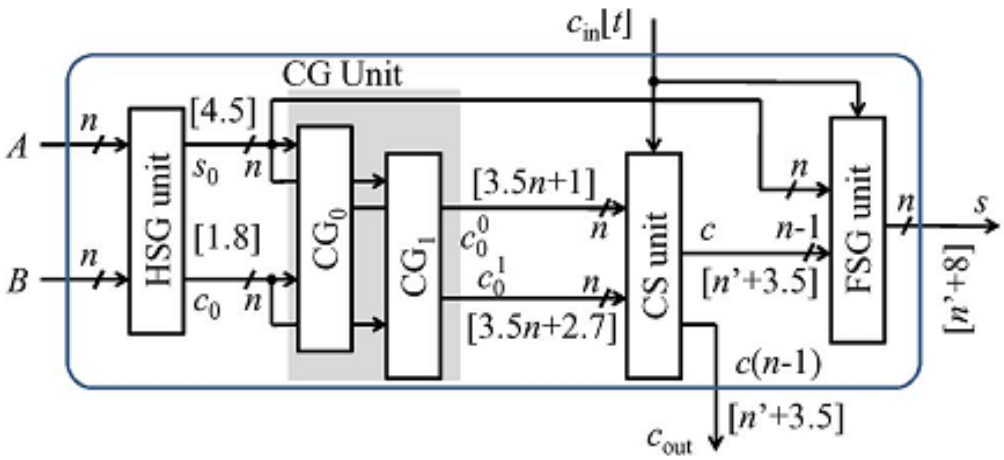

Fig. 5. Proposed CSLA adder for MM Multiplier

In the design shown in fig. 5 the data dependency of the next sum on previous carry has been reduced. Also as the CG0 (Carry Generation with input carry=0) and CG1(Carry Generation with input carry =1) generates the carry independently and then the carry is selected with the help of carry selection unit. Thus the data dependency is optimized.

Montgomery Multiplier With Area Power and Delay Efficient CSLA

The following fig. 6 shows the Montgomery Multiplier which uses the area delay power optimized Carry Select Adder.

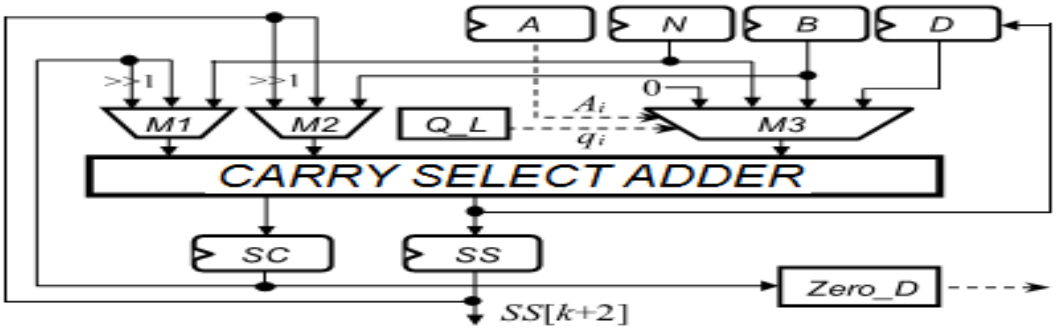

Fig. 6. MM Multiplier with proposed CSLA

From this architecture, on using the newly proposed CSLA, redundant arithmetic operations can be controlled and the data dependency of arithmetic operations can also be enhanced. Even though using this type of carry select adder in MM Multiplier enhances the area required in terms of the number of LUTs required when synthesized under SPARTAN 6 simulation environment in Xilinx ISE, it acquires more delay.

Hence in order to improve in terms of both area and delay we prefer the Carry Skip Adder enhanced in terms of Energy and Delay.

The Carry Skip Adder in general improves the delay performance of adders at the cost of improvement in area usage. The normal Carry Skip Adder (CSKA) is the combination of Full Adders along with the multiplexer which chooses the final Carry. This circuit uses a Carry Skip Logic which is used to detect the situation of all cascaded FAs of CSKA being in propagation

mode and this Carry Skip logic makes the carry ready for the next stage before the calculation of the sum. This Skip operation is done by the circuit which is the combination of logic gates and the multiplexer.

From the normal Conventional CSKA, energy and delay efficient CSKA can be formed by replacing the multiplexer by an incrementation circuit along with the group of AOI circuits. This circuit described above parallel computes the carry and its complement with a small extra hardware with reduced delay [24].

In total it can be said that the newly proposed CSKA is the combination of concatenation and incrementation circuits. As we use the OAI and AOI circuits, the usage of inverter which consumes more power and introduce more delay can be reduced to a great extent. The Fig.7. shows the Proposed Carry Skip Adder with Incrementation Circuit. The MM multiplier which uses the newly proposed CSKA is shown in fig.8.

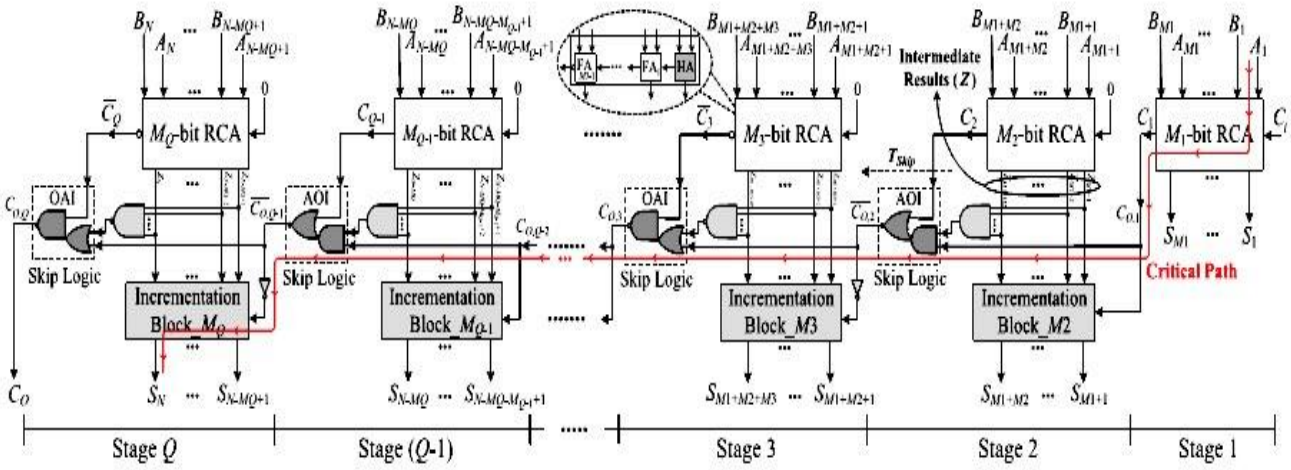

Fig. 7. Proposed Carry Skip Adder with Incrementation Circuit

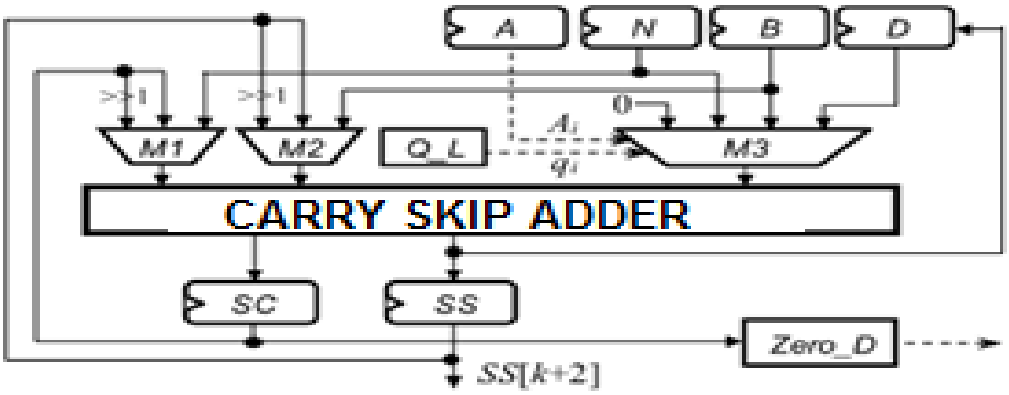

Fig.8. MM Multiplier with proposed incrementation based Carry Skip Adder

IV. Results and Discussion

The above discussed architectures are designed in Verilog Hardware Description Language and are synthesized in Xilinx ISE 14,5 in Spartan 6 kit with . The results have been obtained as in table 1. The table 1 shows the comparison of Montgomery Modular multiplier using various adders and the Comparison charts of Area, Power and Delay are as shown in Fig.9, Fig.10 & Fig. 11. All the synthesis results are taken at the clock frequency of 79.433MHz.

TABLE 1 . SYNTHESIS RESULTS FOR VARIOUS TYPES OF MONTGOMERY MODULAR MULTIPLIER

MMM Type	Area (LUTs)	Power (µW)	Delay (ns)
MMM using CSLA	391	0.208	14.44
MMM using CSKA	406	0.114	7.302

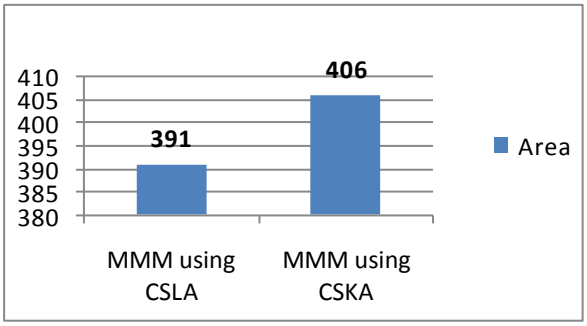

Fig. 9. Area(LUTs) Comparison chart of Adders

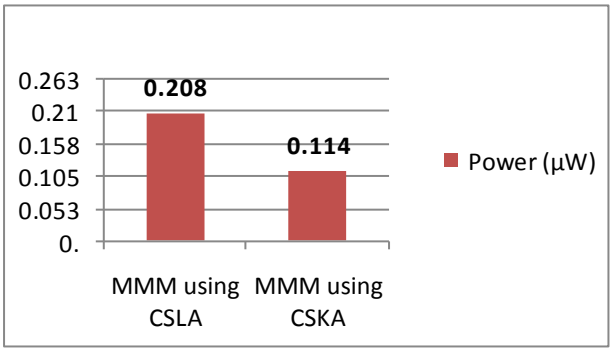

Fig. 10. Power(µW) Comparison chart of Adders

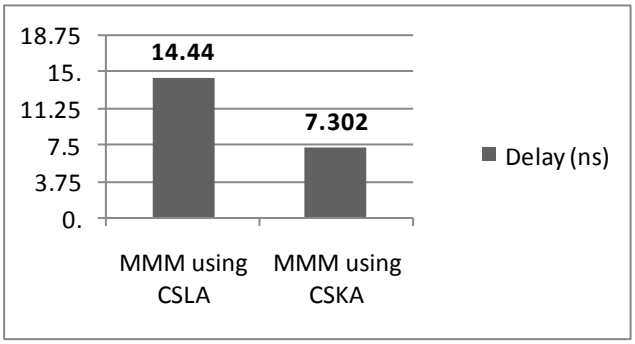

Fig. 11. Delay(ns) Comparison chart of Adders

From the above results it can be seen that the delay of the Montgomery multiplication circuit can be reduced to a great extent to about 50 % by using Carry Skip adder whereas sacrificing a little amount of area overhead.

V. CONCLUSION

The methods namely Carry Select Addition and Carry Skip addition for Montgomery Modular Multiplication has been discussed and their results in terms of area power and delay has been discussed. Hence it is concluded that from the above results, the choice of circuit selection can be based on the requirement needed. In occasions where delay is not of concern, but the area, then CSLA based MM Multiplier can be used. In occasions where area is not of concern but the delay there the faster MM Multiplier with Carry Skip Adder can be used.

References

[1] R. L. Rivest, A. Shamir, and L. Adleman, “A method for obtaining digital signatures and public-key cryptosystems,”Commun. ACM, 21(2), (1978).
[2] V. S. Miller, “Use of elliptic curves in cryptography,” in Advances in Cryptology. Berlin, Germany: Springer-Verlag, (1986).
[3] N. Koblitz, “Elliptic curve cryptosystems,” Math. Comput.,48(177), (1987).
[4] P.L. Montgomery, “Modular multiplication without trial division,”Math. Comput., 44(170), 519–521,(1985).

[5] Y. S. Kim, W. S. Kang, and J. R. Choi, "Asynchronous implementation of 1024-bit modular processor for RSA cryptosystem," inProc. 2nd IEEE Asia-Pacific Conf. ASIC, (2000).

[6] V. Bunimov, M. Schimmler, and B. Tolg, "A complexity-effective version of Montgomery's algorihm," in Proc. Workshop Complex. Effective Designs,(2002).

[7] H. Zhengbing, R. M. Al Shboul, and V. P. Shirochin, "An efficient architecture of 1024-bits cryptoprocessor for RSA cryptosystem based on modified Montgomery's algorithm," inProc. 4th IEEE Int. Workshop Intell. Data Acquisition Adv. Comput. Syst., (2007).

[8] Y.-Y. Zhang, Z. Li, L. Yang, and S.-W. Zhang, "An efficient CSA architecture for Montgomery modular multiplication,"Microprocessors Microsyst., 31(7), (2007).

[9] C. McIvor, M. McLoone, and J. V. McCanny, "Modified Montgomery modular multiplication and RSA exponentiation techniques,"IEE Proc.-Comput. Digit. Techn., 151(6),(2004).

[10] S.-R. Kuang, J.-P. Wang, K.-C. Chang, and H.-W. Hsu, "Energy-efficient high-throughput Montgomery modular multipliers for RSA cryptosystems,"IEEE Trans. Very Large Scale Integr. (VLSI) Syst., 21(11),(2013).

[11] J. C. Neto, A. F. Tenca, and W. V. Ruggiero, "A parallel k-partition method to perform Montgomery multiplication," inProc. IEEE Int. Conf.Appl.-Specific Syst., Archit., Processors, (2011).

[12] J. Han, S. Wang, W. Huang, Z. Yu, and X. Zeng, "Parallelization of radix-2 Montgomery multiplication on multicore platform,"IEEE Trans. Very Large Scale Integr. (VLSI) Syst.,21(12),(2013).

[13] P. Amberg, N. Pinckney, and D. M. Harris, "Parallel high-radix Montgomery multipliers," inProc. 42nd Asilomar Conf. Signals, Syst.,Comput., (2008).

[14] G. Sassaw, C. J. Jimenez, and M. Valencia, "High radix implementation of Montgomery multipliers with CSA," inProc. Int. Conf. Microelectron., (2010).

[15] A. Miyamoto, N. Homma, T. Aoki, and A. Satoh, "Systematic design of RSA processors based on high-radix Montgomery multipliers,"IEEE Trans. Very Large Scale Integr. (VLSI) Syst., 19(7),(2011).

[16] S.-H. Wang, W.-C. Lin, J.-H. Ye, and M.-D. Shieh, "Fast scalable radix-4 Montgomery modular multiplier," in Proc. IEEE Int. Symp. Circuits Syst., (2012).

[17] J.-H. Hong and C.-W. Wu, "Cellular-array modular multiplier for fast RSA public-key cryptosystem based on modified Booth's algorithm,"IEEE Trans. Very Large Scale Integr. (VLSI) Syst.,11(3),(2003).

[18] F. Gang, "Design of modular multiplier based on improved Montgomery algorithm and systolic array," in Proc. 1st Int. Multi-Symp. Comput.Comput. Sci., 2, (2006).

[19] G. Perin, D. G. Mesquita, F. L. Herrmann, and J. B. Martins, "Montgomery modular multiplication on reconfigurable hardware: Fully systolic array vs parallel implementation," in Proc. 6th Southern Program. Logic Conf.,(2010).

[20] A. Cilardo, A. Mazzeo, L. Romano, and G. P. Saggese, "Exploring the design-space for FPGA-based implementation of RSA,"Microprocessors Microsyst., 28(4), (2004).

[21] D. Bayhan, S. B. Ors, and G. Saldamli, "Analyzing and comparing the Montgomery multiplication algorithms for their power consumption," in Proc. Int. Conf. Comput. Eng. Syst., (2010).

[22] C. D. Walter, "Montgomery exponentiation needs no final subtractions," Electron. Lett., 35(21), (1999).

[23] S.R.Kuang, K.Y.Wu and R.Y.Lu, "Low-Cost High-Performance VLSI Architecture for Montgomery Modular Multiplication,"IEEE Trans. Very Large Scale Integr. (VLSI) Syst., 24(2), (2016).

[24] B.K.Mohanty, S.K.Patel, "Area–Delay–Power Efficient Carry-Select Adder,"IEEE Trans,Circuits and Systems-II, 61(6),(2016).