ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 6, Special Issue 5, March 2017

National Conference on Advanced Computing, Communication and Electrical Systems - (NCACCES'17)

24th - 25th March 2017

Organized by

C. H. Mohammed Koya
KMEA Engineering College, Kerala- 683561, India

# A Comparison of Two Algorithms Involving Montgomery Modular Multiplication

Greeshma E[1], Prof. Riyas K S[2]

M.Tech Student, Department of Electronics and Communication Engineering, RIT Kottayam, Kerala, India[1]

Associate Professor, Department of Electronics and Communication Engineering, RIT, Kottayam, Kerala, India[2]

**ABSTRACT:** For future internet services and data communication systems, it is identified that security matters become questionable and problematical. Cryptographic algorithms are a convenient tool for achieving security in those systems. So, realization of cryptographic systems in hardware is more advantageous. Of the two-broad category of cryptographic systems as public key cryptosystems and secret key cryptosystems, public key cryptosystems are widely used. In many public key cryptosystems, the key operation is modular multiplication with large input operands. The trial division in modular multiplication is time consuming. So, well-known algorithm called Montgomery modular multiplication algorithm is introduced by avoiding the trial division. Shifting modular additions are used instead of complicated division operations. Different modifications to conventional Montgomery modular multiplications are proposed to reduce the delay associated with the long carry propagation in the computation of intermediate result. This paper explores a comparison between two modification algorithms to conventional Montgomery MM algorithms.

**KEYWORDS:** Cryptography, Montgomery modular multiplication, Public-key cryptosystems

## I. INTRODUCTION

Several open network services such as electronic commerce, electronic shop for music, video is expected to make life more secure and helpful. To realize these networks, encryption technologies are used. A broad classification of these encryption technologies are public key cryptosystems and secret key cryptosystems. Of these, Public key cryptosystems are widely used because of the use of entirely different keys for encryption and decryption. So, it is trustworthy and provides data integrity. The reduction technique in most of the public key cryptosystems is modular multiplication as described in [2].

For modular multiplication without trial division an algorithm is proposed by P. L. Montgomery in 1985. The algorithm incorporates a series of shifting modular additions to create final result. After the introduction of Montgomery MM algorithm, a number of algorithms are presented to speed up above mentioned algorithm. These algorithms come under the broad category of either SCS or FCS strategy. SCS strategy receives and outputs data in binary representation but the intermediate result S [i+1] is in carry save format. So, in the final step, a conversion of format from carry save to binary is required. For this conversion, a carry propagation adder or iterative use of carry save adder can be adopted. In FCS strategy, all the operand values are kept in carry save format so no need of such format conversion. Even though FCS strategy reduces the number of clock cycles needed for completing one modular multiplication, it increases the hardware complexity and critical path of entire system.

One of the main reasons associated with the delay is long carry propagation in the calculation of intermediate result as shown in step 4 of conventional Montgomery MM algorithm. For reducing the carry propagation time, many approaches based on carry save strategies are introduced.

The two main advancements in the field are conversion of addition operation in to one selection operation in the calculation of intermediate result and the use of a Configurable Carry Save Adder for addition, format conversion and operand pre-computation. i.e., the CCSA architecture is used iteratively for format conversion from carry save format to binary. The selection operation is carried out in such a way that the operand 0, N, B or D is selected depending on the select lines Ai and qi of a 4 X 1 multiplexer. This CCSA architecture can be switched to one full adder and two serial half adders depending on certain parameters. This paper present a comparison between the two-works mentioned above.

Table.1: Algorithm for Radix-2 Montgomery modular multiplication

*Inputs* : $A$, $B$, $N$ (modulus)
*Output* : $S[k]$

1. $S[0] = 0$;
2. for $i = 0$ to $k - 1$ {
3. $q_i = ( S[i]_0 + A_i \times B_0 ) \bmod 2$;
4. $S[i+1] = ( S[i] + A_i \times B + q_i \times N ) / 2$;
5. }
6. if ( $S[k] \geq N$ ) $S[k] = S[k] - N$;
7. return $S[k]$;

The two main advancements in the field are conversion of addition operation in to one selection operation in the calculation of intermediate result and the use of a Configurable Carry Save Adder for addition, format conversion and operand pre-computation. i.e., the CCSA architecture is used iteratively for format conversion from carry save format to binary. The selection operation is carried out in such a way that the operand 0, N, B or D is selected depending on the select lines Ai and qi of a 4 X 1 multiplexer. This CCSA architecture can be switched to one full adder and two serial half adders depending on certain parameters. This paper present a comparison between the two-works mentioned above.

A FCS-MMM42 multiplier with the specialty of high energy efficiency is proposed by Kuang et al. [13] in which energy dissipation is reduced by destroying unnecessary operations of the four to two CSA architecture thus the throughput is enhanced. However, higher area and critical path delay are the main problems of FCS-MM42 multiplier. A complexity effective version of Montgomery MM is introduced in [5]. Modified Montgomery modular multiplication and RSA exponentiation techniques are explained in [14]. For increasing the speed of conventional Montgomery MM algorithm, techniques of parallelism, high radix algorithms instead of radix 2, concept of systolic array design are used. These architectures suffer from the problem of higher power dissipation.

The reminder of this paper is organized as follows. Section II briefly reviews the method of conventional Montgomery modular multiplication. In Section III, modified Montgomery modular multiplication by using selection operation is described. Section IV describes about modified Montgomery modular multiplication by using CCSA architecture. Results are given in section V. Finally, conclusion is given in section VI.

## II. MONTGOMERY MODULAR MULTIPLICATION

Montgomery algorithm defines the quotient qi of modular multiplication only depending on the LSB of operands and thus complicated division in conventional MM is replaced with a series of shifting modular additions to produce S= A X B $X$ $R^{-1}$ (mod N), where N is modulus value, $R^{-1}$ is the inverse of R modulo N and R=$2^k$ mod N as described in [3]. Montgomery modular multiplication is an easier algorithm for modular multiplication without trial division. In modular multiplication, two integers A, B and a modulus value, N is given; the problem is to find AXB mod N. A block diagram representation of conventional Montgomery MM is given in Fig.1.

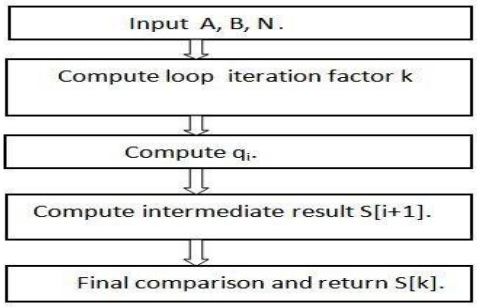


Fig. 1: Block diagram representation of conventional MM

Here, the input operands A, B and modulus value N is given in binary representation. The operands are changed to Montgomery form. i.e., AX R mod N and BXR mod N for A and B respectively. Then, the next step computes loop

iteration factor k. The value of k can be calculated by using the relation $R=2^k$ mod N where R is the radix and N is the modulus value. Next step computes quotient of Montgomery MM by using the relation qi = LSB of S[i]+(Ai*B0)where Ai is the $i^{th}$ bit of operand 'A' and B0 is the LSB of operand 'B'. The next step is the computation of intermediate result S[i+1] by using S[i+1]=S[i]+(Ai*B + qi*N). The loop is iterated k times and the result S[k] is obtained. Since the convergence range of S[k] is in between 0 and 2N, a final comparison and subtraction is needed. i.e., if S[k] is greater than N, N is subtracted from S[k] and the final result is returned. This comparison is avoided in a work by C D Walter [4] by changing the number of iterations from k to k+2. i.e., $R=2^{k+2}$ mod N.

### *A. SCS BASED MONTGOMERY MM*

SCS (Semi Carry save Strategy) as the name indicates, the only value that kept in carry save format is intermediate result S as SS and SC to avoid carry propagation. Other operands are kept in binary representation. But the values kept in carry save format are needed to convert to binary representation. For this conversion, additional hardware is required. In [9], Carry propagation adder is used at the final step of MM. In paper [12], the conversion is made by using a carry save architecture repeatedly. i.e., SS and SC are added until SC=0. Fig 2 shows architecture of SCS based Montgomery multiplier

### *B. FCS BASED MONTGOMERY MM*

Here, all operands A, B and S are kept in carry save format as (AS, AC), (BS, BC) and (SS, SC). Since all operands are kept in carry save format, no need of final format conversion hence lower number of clock cycles is required to complete one Montgomery modular multiplication. But the hardware complexity and critical path is increased. The advantage of FCS strategy is, it requires lesser clock cycles to complete one modular multiplication. Two FCS based Montgomery multipliers, denoted as FCS-MM-1 and FCS-MM-2 multipliers are proposed by McIvor et al. [13]. FCS-MM-1 composed of one five-to-two and FCS-MM-2 composed of one four-to-two Carry Save Adder architectures. Figure 3 shows a simplified architecture of FCS-MM1 multiplier. It consists of two shift registers, a full adder (FA), and a flip-flop (FF).

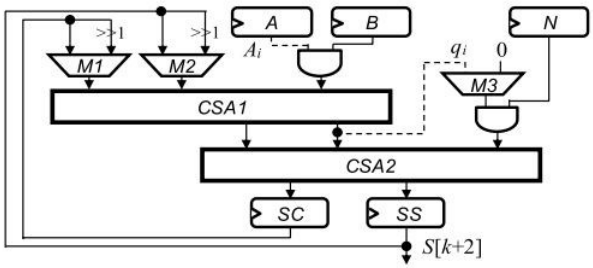


Fig. 2: SCS based Montgomery multiplier architecture

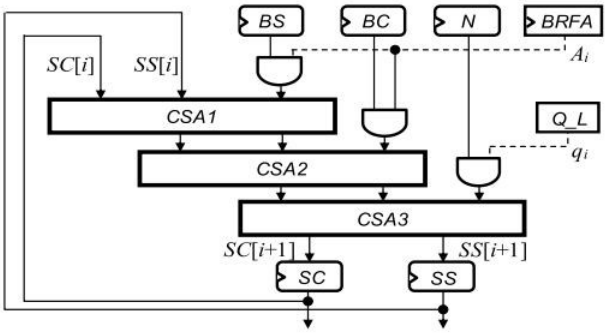


Fig. 3: FCS-MM1 Architecture

## III. MONTGOMERY MM BY USING THE CONCEPT OF SELECTION OPERATION

The main contribution to delay is long carry propagation in the calculation of intermediate result S [i+1] =S[i] + (Ai*B + qi*N) this addition operation can be changed to one selection operation using one 4 X 1 multiplexer. The selection is in such a way that, if Ai=0 and qi=0 then x=0, if Ai=0 and qi=1 then x=n, if Ai=1 and qi=0 then x=B and if Ai=1 and qi=1 then x=D provided S [i+1] =(S[i] +x)/2 where D=B+N. Figure 4 below shows the architecture.

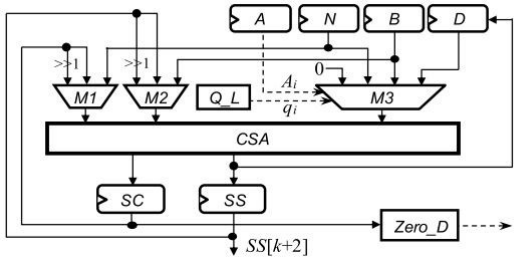


Fig. 4: modified MM architecture using selection operation

## IV. MONTGOMERY MM BY USING CCSA ARCHITECTURE

The use of carry save architecture for addition, format conversion, operand pre-computation will reduce the critical path. But one problem is extra clock cycles are needed to complete one modular multiplication. To reduce the clock cycle number, a new architecture is called Configurable Carry Save Adder is used in [1]. A CCSA architecture is a structure which can be used as a full adder or two serial half adders depending on the value as shown in figure 5. Also for simplicity, normal 4X1 multiplexer is replaced with another multiplexer SM3 as in figure 6.

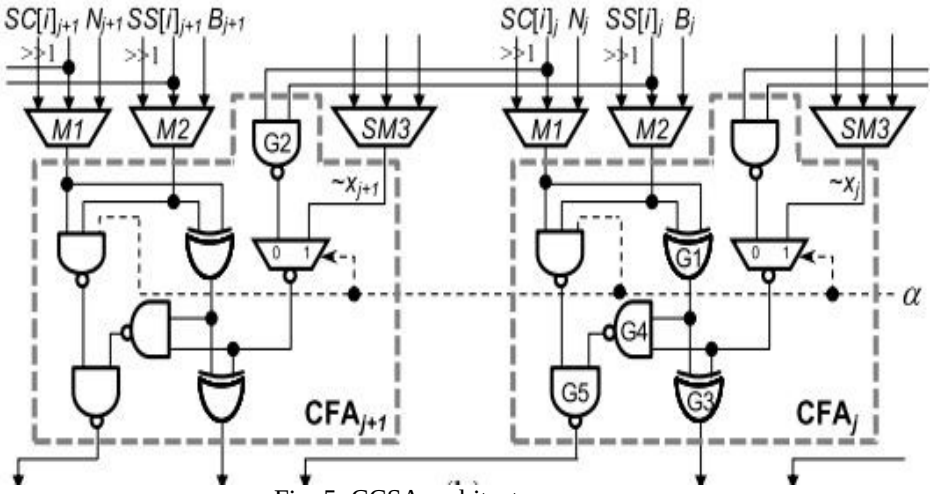


Fig. 5: CCSA architecture

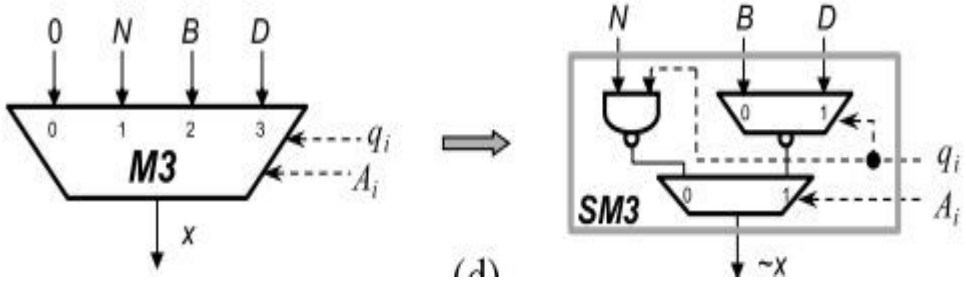


Fig. 6: Architecture of Multiplexer SM3

If $\alpha$ =1, CCSA act as one FA and three input carry save addition is performed. Otherwise, it acts as serially connected two half-adders (HAs) and it can be used to perform two serial two input carry save additions.

## V. RESULTS

This section, analyze and compares the delay and area of above mentioned two designs. The results summarized in Table. 2.

TABLE.2: Result comparison of two Montgomery multipliers

| Montgomery multiplied | Delay | Area | Time | Throughput | ATP |
|---|---|---|---|---|---|
| Using selection operation | 5.6 | 406127 | 5.8744 | 174.3 | 2385.75 |
| Using CCSA architecture | 4.0 | 498379 | 3.5200 | 290.9 | 1754.29 |

## VI. CONCLUSION

Montgomery modular multiplication algorithm is an algorithm developed to achieve modular multiplication without trial division. To reduce long carry propagation, modification works adopt either SCS or FCS method. Both SCS and FCS strategies have their own advantages and disadvantages. In SCS strategy, since intermediate result is kept in carry save format, in the final step of Montgomery modular multiplication, a format conversion from carry save format to binary format is needed. So, extra hardware for this will increase the critical path and hardware cost of entire multiplier system. In FCS system, all the operands are kept in carry save format like (AS, AC), (BS, BC), (SS, SC). This avoids the format conversion at the final step. But the number of clock cycles needed are larger compared with SCS strategy. This paper made a comparison between two algorithms presented in the literature as a modified work of conventional Montgomery modular multiplication algorithm. The first described algorithm is a modified work of conventional Montgomery multiplication algorithm that uses a 4 X 1 multiplier for selection of operands that replaces long carry propagation addition. In the second algorithm that uses a single Configurable Carry Save Adder for addition, operand pre-computation, format conversion from carry save format to binary format. The CCSA is an adder architecture that performs either full addition of three operands or half addition of two operands depending upon certain parameters. Also, a special multiplexer architecture is used in the work to make the system simple. Experimental results show that the algorithm that uses CCSA architecture performs better in terms of throughput and Area Time Product (ATP).

## REFERENCES


[1] Shiann-Rong Kuang,Kun-Yi Wu, and Ren-Yao Lu "Low-Cost High-Performance VLSI Architecture for Montgomery Modular
[2] Multiplication," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 24, NO. 2, February 2016
[3] Diffie and Hellman"A method for obtaining digital signatures and public-key cryptosystems,," Commun. ACM, vol. 21, no. 2, , Feb. 1978.
[4] P.L. Montgomery, "Modular multiplication without trial division," Math. Comput, vol. 44, no. 170, pp.519−521, Apr. 1985.
[5] C. D. Walter, "Montgomery exponentiation needs no final subtractions,"Electron. Lett, vol. 35, no. 21, Oct. 1999.
[6] V. Bunimov, M. Schimmler, and B. Tolg, "A complexity-effective version of Montgomerys algorithm, "in Proc. Workshop Complex. ffective Designs, May 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 Montgomerys algorithm,"*inProc. 4th IEEE Int. Workshop Intell. Data Acquisition Adv. Comput. Syst., Sep. 2007
[8] Elbirt, A.J., and Paar, C.:"Towards an FPGA Architecture Optimized for Public-Key Algorithms,"Presented at the SPIE Symp. on Voice, Video and Communications, Sept. 1999
[9] Kimet et.al:"*Exploring the design space for FPGA based implementation of RSA,"*Microprocessors Microsyst., vol. 28, no. 4, pp. 183191, May 2004.
[10] 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, Aug. 2000.
[11] 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., Nov. 2010, pp. 257261
[12] 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., Mar. 2010
[13] Y.-Y. Zhang, Z. Li, L. Yang, and S.-W. Zhang, "An efficient CSA architecture for Montgomery modular multiplication,"Microprocessors Microsyst., vol. 31, no. 7, Nov. 2007
[14] S.-R. Kuang, J.-P. Wang, K.-C. Chang, and H.-W. Hsu, "Energy-efficient high-throughput Montgomery modular multipliers for RSA cryptosys-tems,"IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 21, no. 11
[15] C. McIvor, M. McLoone, and J. V. McCanny, "Modified Montgomery modular multiplication and RSA exponentiation techniques,"IEE Proc.-Comput. Digit. Techn., vol. 151, no. 6, Nov. 2004.