IJIREEICE

ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

nCORETech 2017

LBS College of Engineering, Kasaragod

Vol. 5, Special Issue 1, March 2017

VLSI Architecture for High Performance Montgomery Modular Multiplication

Khadeeja Thameema[1], **Santo Mathew**[2]

Student, M.Tech, VLSI Design & Signal Processing, LBS College of Engineering Kasaragod[1]

Assistant Professor, ECE Department, LBS College of Engineering Kasaragod[2]

Abstract: Montgomery Modular Multiplication is a method for performing fast modular multiplication. Montgomery Modular Multiplication is used for encryption process in Public Key Cryptography.This paper proposes a Semi Carry Save (SCS) based Montgomery Modular Multiplication, with high speed performance. One Carry Save Adder(CSA) is used to avoid carry propagation at each addition operation. The CSAis reused for operand pre-computation and format conversion leading to a short critical path delay. Proposed Montgomery modular Multiplier is implemented using VHDL language in Xilinx software. The proposed multiplier is then compared with one of the existing Montgomery multiplier in terms of area, delay and power.

Keywords: Carry Save Adder, Montgomery Modular Multiplication, Semi Carry Save , Public Key Cryptography.

I. INTRODUCTION

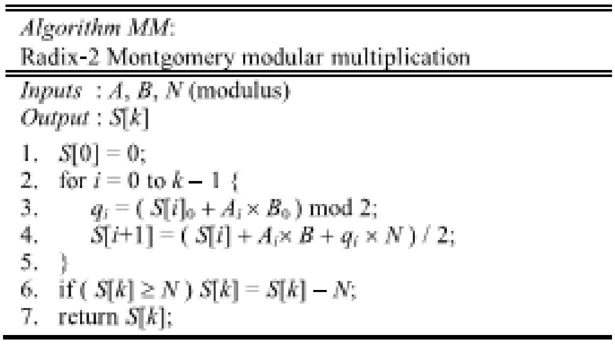
Modular Multiplication (MM) with large integers is a time consuming operation in many public-key cryptosystems [1]. Therefore, many algorithms have been presented to carry out MM more quickly and Montgomery's algorithm is one of them. Montgomery's algorithm determines the quotient only depending on the least significant digit of operands [2]. It replaces the complicated division in MM with a series of shifting modular additions.

Montgomery algorithm is classified into two based on the representation of input and output operands. They are Full-Carry-Save Montgomery modular Multiplication (FCS-MM) and Semi-Carry-Save Montgomery modular Multiplication (SCS-MM). In FCS-MM both the obtained sum and carry are considered as output. But in SCS-MM only the obtained sum is considered as output. The adder levels in SCS-MM is less. Therefore SCS-MM requires lower area than FCS-MM. Hence SCS based multiplier is modified here.

The remainder of this paper is organised as follows. Section II briefly describes about Montgomery MM algorithm. Section III briefly reviews the existing SCS based Montgomery multipliers. Section IV describes the proposed SCS based Montgomery multiplier. The comparisons of existing and proposed multipliers are made in Section V. The conclusion is drawn in Section VI.

II. MONTGOMERY MM ALGORITHM

The Montgomery modular product S of A and B can be obtained as S = A × B × R^{-1} (mod N), where R^{-1} is the inverse of R modulo N. That is, R × R^{-1} = 1 (mod N). The length of A,B and N should be same. Also the value of N should be greater than A and B.



Algorithm MM:
Radix-2 Montgomery modular multiplication

```
Inputs  : $A$, $B$, $N$ (modulus)
Output : $S[k]$
$S[0] = 0$;
for $i = 0$ to $k - 1$ {
    $q_i = ( S[i]_0 + A_i \times B_0 ) \bmod 2$;
    $S[i+1] = ( S[i] + A_i \times B + q_i \times N ) / 2$;
}
if ( $S[k] \geq N$ ) $S[k] = S[k] - N$;
return $S[k]$;
```

Fig. 1 Montgomery MM Algorithm

Fig 1. shows Montgomery MM Algorithm. K represents the number of bits and i represents i^{th} bit. If the final product S is greater than N, then N is subtracted from S which is shown in step 7 of Fig 1.

III. EXISTING SCS BASED MULTIPLIERS

The Montgomery modular product S of A and B is obtained as S = A × B × R^{-1} (mod N), where R^{-1} is the inverse of R modulo N. That is, R × R^{-1} = 1 (mod N). The intermediate result S of shifting modular addition is kept in the carry-save representation (SS, SC) to avoid long carry propagation. The format conversion from the carry-save format of the final modular product into its binary format is needed. The two existing SCS based Montgomery Multiplier are SCS-MM1 and SCS- MM2. Fig 2 shows the architecture of SCS-based MM algorithm proposed in [3] (denoted as SCS-MM1 multiplier) ,composed of twoCarry Save Adders (CSA) architecture and one format converter, Carry Propagation Adder (CPA), where the dashed line denotes a 1-bit signal.

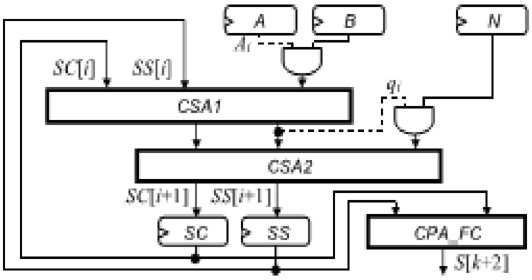


Fig. 2 SCS-MM1 multiplier

The extra CPA enlarges the area and the critical path of the SCS-MM-1 multiplier.

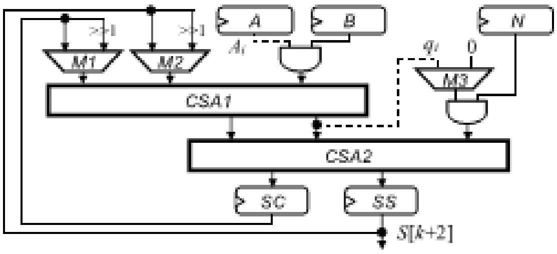


Fig. 3 SCS-MM2 multiplier

SCS MM 1 is modified by reusing the two-level CSA architecture for performing the format conversion so that the CPA can be removed. Fig. 3 shows the architecture of the Montgomery multiplier proposed in [4] (denoted as SCS-MM-2 multiplier). This multiplier is modified further to reduce the critical path delay and area to increase the performance.

IV. PROPOSED SCS BASED MULTIPLIER

The critical path delay of SCS-based multiplier is reduced by pre-computing D = B + N . Two CSA's are replaced by one CSA [6]. The CSA is reused for performing B+ N and the format conversion. Fig.3 shows the hardware architecture of modified SCS-based Montgomery multiplier (MSCS-MM) . The Zero_D circuit in Fig. 3 is used to detect whether SC is equal to zero, which can be accomplished using one NOR operation.

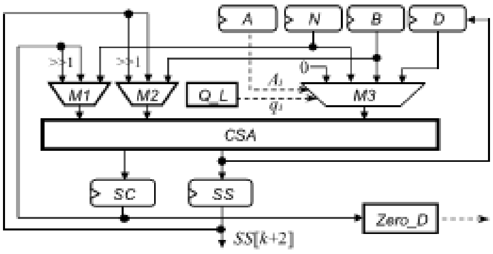


Fig.4 MSCS-MM multiplier

The carry propagation addition operations of B + N and the format conversion are performed by the one-level CSA architecture of the MSCS-MM multiplier through repeatedly executing the carry-save addition. Therefore, the critical path delay of the MSCS-MM multiplier can be reduced.

The area complexity is also reduced as only one level CSA is used here.The structure of carry save adder used is shown in Fig 5. The CSA block internally consists of full adders which is realized using and gates and xor gates.

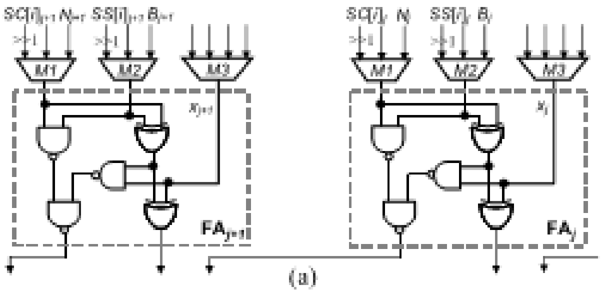


Fig. 5 Two cells of CSA

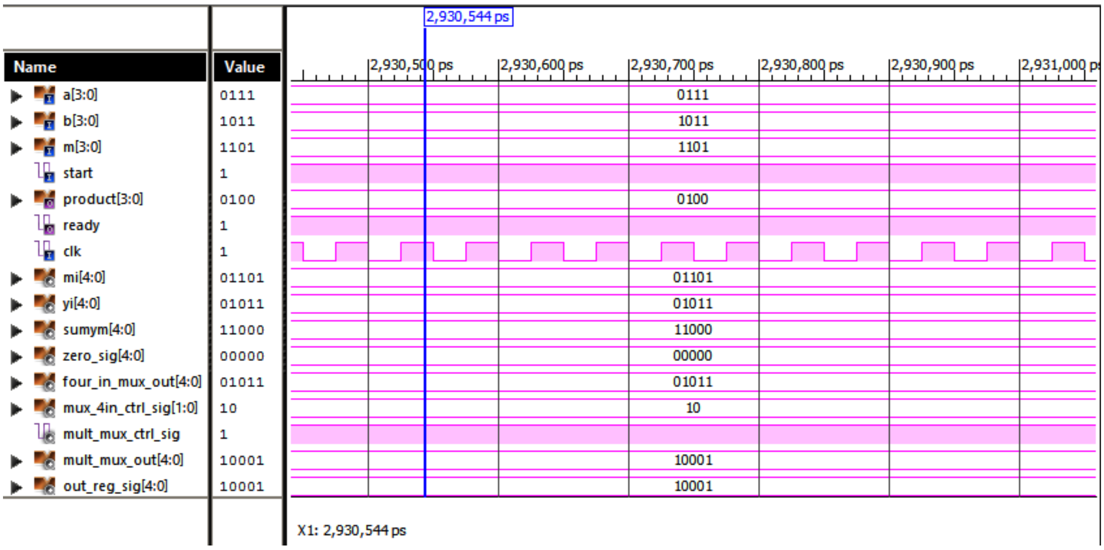
The carry save adder is used because it have less propagation delay. Carry Save adder for n-bit means it have n-parallel adders, which produce n-bit sums and n-bit carry's. The inputs for carry save adder are SS,SC and mux output.

V. EXPERIMENTAL RESULTS

The design of Modified SCS-MM (MSCS-MM) has been made by using VHDL. The simulation results have been evaluated by using Modelsim 10.3c and synthesis Performances are estimated by using Xilinx 14.7 , for 4-bit and 8-bit. The simulation results are shown in Fig 6 and Fig 7.The critical path delay , area and power of the proposed multiplier is analyzed . This is then compared with the area , delay and power of SCS MM-2. The delay ,area and power of the proposed multiplier have been decreased. Therefore the speed the proposed multiplier is increased.The results are shown in Table I.

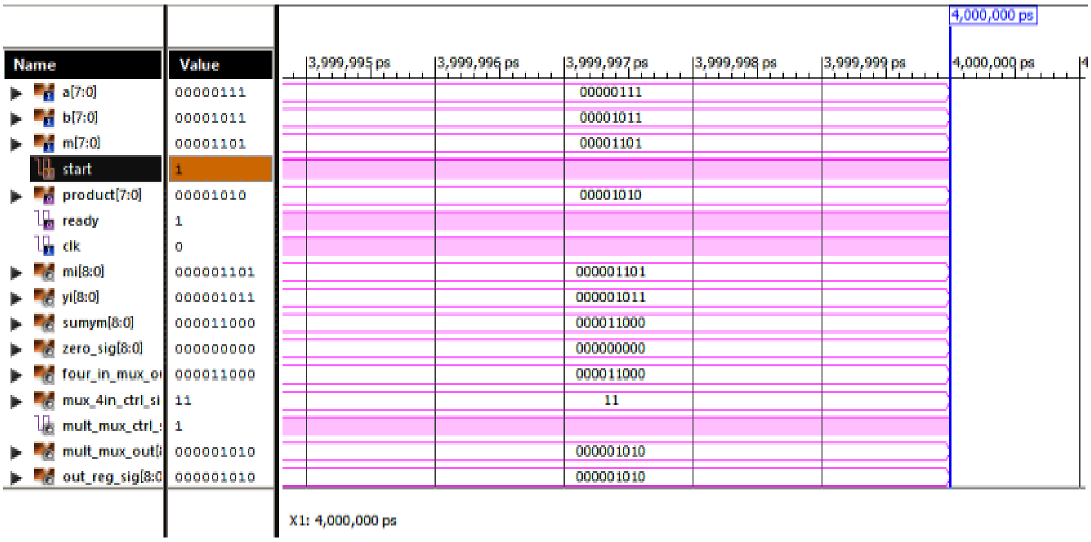
TABLE I ANALYSIS OF SCS-MM2 AND MODIFIED SCS-MM

No.of Bits	Multiplier	Delay(ns)	Cell Area	Switching Power(nW)
4	SCS MM-2	13.63	2318	105924.57
	MSCS MM	**9.649**	**1347**	**95348.23**
8	SCS MM-2	23.81	4217.88	198262.003
	MSCS MM	**15.792**	**2651.4**	**113348.021**



Name	Value
a[3:0]	0111
b[3:0]	1011
m[3:0]	1101
start	1
product[3:0]	0100
ready	1
clk	1
mi[4:0]	01101
yi[4:0]	01011
sumym[4:0]	11000
zero_sig[4:0]	00000
four_in_mux_out[4:0]	01011
mux_4in_ctrl_sig[1:0]	10
mult_mux_ctrl_sig	1
mult_mux_out[4:0]	10001
out_reg_sig[4:0]	10001

Fig 6 Simulation Waveform of 4 bit MSCS MM algorithm



Name	Value
a[7:0]	00000111
b[7:0]	00001011
m[7:0]	00001101
start	1
product[7:0]	00001010
ready	1
clk	0
mi[8:0]	000001101
yi[8:0]	000001011
sumym[8:0]	000011000
zero_sig[8:0]	000000000
four_in_mux_o	000011000
mux_4in_ctrl_si	11
mult_mux_ctrl_:	1
mult_mux_out[	000001010
out_reg_sig[8:0	000001010

Fig 6 Simulation Waveform of 4 bit MSCS MM algorithm

VI. CONCLUSION

In the Modified SCS based Montgomery Multiplier (MSCS MM) the critical path delay,area and switching power is reduced,when compared to the existing logic. This multiplier used only one carry save adder. Thus the proposed architecture can achieve reduced critical path, and increase the speed of operation.

REFERENCES

[1] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," Commun. ACM, vol. 21, no. 2, pp. 120–126, Feb. 1978.

[2] P. L. Montgomery, "Modular multiplication without trial division," Math.Comput ., vol. 44, no. 170, pp. 519–521, Apr. 1985.

[3] Y. S. Kim, W. S. Kang, and J. R. Choi, "Asynchronous Implementation of 1024-bit modular processor for RSA cryptosystem," in Proc. 2nd IEEE Asia-Pacific Conf. ASIC, Aug. 2000, pp. 187–190.

[4] Y.-Y. Zhang, Z. Li, L. Yang, and S.-W. Zhang, "An efficient CSA Architecture for Montgomery modular multiplication," MicroprocessorsMicrosyst., vol. 31, no. 7, pp. 456–459, Nov. 2007.

[5] Harmeet Kaur, Mrs.CharuMadhu,"Montgomery Multiplication Methods - A Review" International Journal OfApplication Or Innovation In Engineering & Management (Ijaiem) Volume 2, Issue 2, February 2013.

[6] Kuang et.al. "Low Cost High Performance VLSI Architecture for Montgomery Modular Multiplication", IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 24, issue: 2,pp434-445 , Feb.2016.