

Modified Montgomery Modular Multiplication Using 4:2 Compressor and CSA Adder

Himanshu Thapliyal, Anvesh Ramasahayam*, Vivek Reddy Kotha*,
Kunul Gottimukkula* and M.B Srinivas

Centre for VLSI and Embedded System Technologies

International Institute of Information Technology, Hyderabad, 500019, India

**Department of Electronics and Communication Engineering,*

S.R. Engineering College, Warangal, India

(thapliyalhimanshu@yahoo.com, srinivas@iiit.net)

Abstract

The efficiency of the Public Key encryption systems like RSA and ECC can be improved with the adoption of a faster multiplication scheme. In this paper, Modified Montgomery multiplications and circuit architectures are presented. The first modified Montgomery multiplier uses 4:2 compressor and carry save adders (CSA) to perform large word length additions. The total delay for a single modular multiplication using the proposed approach is $7XOR+1$ AND gate compared to $8XOR+1$ AND gate of the recently proposed fastest algorithm. The second modified Montgomery multiplier uses a novel proposed hardware unit that outputs carry save representation of the 4-input operands in $3XOR$ delays. The total delay for a single modular multiplication using the novel hardware unit is $5XOR+1$ AND gate compared to $6XOR+1$ AND gate of the recently proposed algorithm. The optimal transistor implementations of the proposed approaches have also been presented. The proposed transistor implementations are highly optimized in terms of area, speed and low power. The proposed Montgomery multiplication circuit will be of eminent importance when implemented for higher word length such as 1024 and 2048 as there will be saving in the propagation delays by 1024 and 2048 XOR gates respectively compared to the recently proposed fastest algorithm.

1. Introduction

RSA and ECC (Elliptic Curve Arithmetic) are the two major standards used for public-key cryptography. Modular multiplication is the major time consuming operation in the public key cryptography algorithms like RSA and ECC. Among

the modular multiplication available, the Montgomery multiplication algorithm [1] is the most efficient scheme of calculating $A*B \text{ Mod } N$. It facilitates the replacement of division operation by the modulus N with a series of additions and divisions by a power of two. Thus, Montgomery multiplication satisfies the constraint of VLSI design by being fast on the one hand and being area and power efficient on the other hand. There are several techniques proposed for further improving the hardware implementation of the Montgomery multiplication.

Recently, two modified Montgomery multiplication algorithms suitable for RSA exponentiation are proposed better than the existing approaches [5] which avoid these problems. The modifications were based on using a five-to-two CSA (three levels of CSL (Carry Save Logic)) and a four-to-two CSA (two levels of CSL), respectively. The approaches proposed in [5] can perform Montgomery multiplication in only $k+1$ and $k+2$ clock cycles, respectively, where k is the operand bit length. They use all the inputs and outputs to the Montgomery multipliers in a redundant carry save format. This paper proposes two approaches to improve modified Montgomery multiplication proposed in [5], in terms of critical delay. The proposed approaches are also based on the five-to-two CSA and four-to-two CSA logics, but implements the novel hardware units for reducing the levels as well as the critical delays. The hardware units proposed for improving the algorithms of [5] are discussed in the later sections of the paper. The optimal transistor implementations of the proposed approaches of performing five-to-two CSA and four-to-two CSA operations are also been presented.

2. Montgomery Multiplication algorithms

Peter Montgomery [1] has provided an algorithm called Montgomery's modular multiplication algorithm for computing the modular multiplication. The major contributing factor in the critical delay is the carry propagation resulting from the very large operand additions. In order, to improve the critical delay, two modified version of algorithms are proposed in [5]. The first algorithm proposed in [5] is shown below.

Algorithm 1: Five-to-two CSA Montgomery Multiplication proposed in [5]

```
(A1, A2, B1, B2, n)
S1[0]=0; S2[0]=0;
For i in 0 to k-1 loop
  qi=(S1[i]0+S2[i]0)+(Ai * (B10+B20)) mod 2;
  S1[i+1],S2[i+1]=CSR(S1[i]+S2[i]+Ai*(B1+B2)+qi * n)
  div 2;
end loop;
return S1[k],S2[k];
```

The first algorithm is based on a five-to-two CSA. In the above algorithm, the input operands A and B and the output product S are in a carry save representation (CSR denoted by A1 and A2, B1 and B2, and S1 and S2 respectively). In the Algorithm 1, the three levels of CSL (Carry Save logic) are used in five-to-two CSA. The critical delay of algorithm 1 occurs during the calculation of the five-to-two carry save addition [5]:

$$S1[i+1], S2[i+1] = CSR(S1[i] + S2[i] + A_i(B1+B2) + q_i, n) \dots (1)$$

Using the approach of five-to-two CSA proposed in [5], the critical delay of the algorithm 1 will be

$$3 \text{ Full Adders} + 2 \text{ XORs} + 1 \text{ AND}$$

$$= 3 * 2 \text{ XOR} + 2 \text{ XOR} + 1 \text{ AND}$$

$$= 8 \text{ XOR} + 1 \text{ AND, since the delay of the full adder is } 2 \text{ XOR gates.}$$

The extra 2 XOR gates and 1 AND gate in the critical delay of the algorithms 1 occur due to the computation of q_i in line 1 of the for loop. The second algorithm proposed in [5] is based on four-to-two CSA rather than five-to-two CSA resulting in a saving of a full level of CSL. The modified algorithm 2 is shown below.

Algorithm 2: Four-to-two CSA Montgomery Multiplication proposed in [5]

```
(A1, A2, B1, B2, n)
D1, D2 = CSR(B1 + B2 + n + 0)
S1[0]=0; S2[0]=0;
For i in 0 to k-1 loop
  qi=(S1[i]0+S2[i]0)+(Ai * (B10+B20)) mod 2;
  if Ai=0 and qi=0 then
    S1[i+1],S2[i+1]=CSR(S1[i]+S2[i]+0+0) div 2;
  elseif Ai=1 and qi=0 then
    S1[i+1],S2[i+1]=CSR(S1[i]+S2[i]+B1+B2) div 2;
  elseif Ai=0 and qi=1 then
    S1[i+1],S2[i+1]=CSR(S1[i]+S2[i]+ n+0) div 2;
  else
    S1[i+1],S2[i+1]=CSR(S1[i]+S2[i]+D1+D2) div 2;
  endif;
endloop;
return S1[k],S2[k];
```

Four-to-two carry save addition is the prominent operation in the Algorithm 2 of [5]. It can be represented as

$$S1[i+1], S2[i+1] = CSR(S1[i] + S2[i] + y + z)$$

where the pair y and z values depend on A_i and q_i and depending on values can represent either zero and zero, B1 and B2, n and zero or D1 and D2 respectively [5]. In order to determine the state of these signals simultaneously, two 4:1 multiplexer working in parallel are required. Hence, the delay of 4:1 MUX also needs to be taken into account in determining the critical delay. Using the four-to-two approach, the critical delay of algorithm 2 for single modular multiplication is

$$2 \text{ Full Adders} + 4:1 \text{ Multiplexer} + 2 \text{ XORs} + 1 \text{ AND}$$

$$= 4 \text{ XOR} + 4:1 \text{ Multiplexer} + 2 \text{ XORs} + 1 \text{ AND}$$

$$= 6 \text{ XOR} + 4:1 \text{ MUX} + 1 \text{ AND Gate.}$$

Table 1. Timing Delays Required for a Single Modular Multiplication When used for Exponentiation Operation in Cryptosystems

Algorithm	Loop Delay	Conversion Delay	Clock Cycles	Critical Delay
Algorithm1[5]	3CSA+1XORs+1AND	None	K+1	3 full Adders +2XORs+1AND =8XORs+1AND
Proposed Improvement in Algorithm 1	1 4:2 Compressor+1 CSA+2XORs+1AND	None	K+1	1 4:2 Compressor+1 CSA+2XORs+1AND =3XORs+2XORs+2XORs+1AND =7XORs+1AND
Algorithm 2[5]	2 CSA+4:1 Multiplexer+2XORs+1AND	None	K+2	2 Full Adders+4:1 Multiplexer+2XORs+1AND =4XORs+4:1 Multiplexer+2XORs+1AND =6XORs+4:1 Multiplexer+1AND
Proposed Improvement in Algorithm 2	1 Novel Hardware Unit+4:1 Multiplexer+2XORs+1AND	None	K+2	1 Novel Hardware Unit+4:1 Multiplexer+2XORs+1AND =3XORs+4:1 Multiplexer+2XORs+1AND =5XORs+4:1 Multiplexer+1AND

3. Proposed Approach of improvement

The proposed approach uses 4:2 compressor for the improvement of Algorithm 1 in [5] and a novel hardware unit for the improvement of the Algorithm 2 in [5].

3.1 Improvement of Algorithm 1 proposed in [5]

The Algorithm 1 in [5] was based on a five-to-two CSA having three levels of CSL and is used to calculate (1). The algorithm critical delay is highly dependent on the three levels of CSL. The proposed approach introduces the use of two levels instead of three levels for the computation of five-to-two CSA. In the proposed approach, the computation is reduced to two levels through the use of 4:2 compressor at the first level and then feeding its output to the full adder to get the output at the second level. The proposed approach is shown in Figure 1. There is a significant improvement in the delay using the proposed approach as the propagation delay of the 4:2 compressor is 3XOR gates [2]. Hence, the total delay for a single modular multiplication using the proposed approach is

$$1 \text{ 4:2 Compressor} + 1 \text{ Full Adder} + 2 \text{ XORs} + 1 \text{ AND} \\ = 3 \text{ XOR} + 2 \text{ XOR} + 2 \text{ XOR} + 1 \text{ AND} \\ = 7 \text{ XOR} + 1 \text{ AND Gate}$$

The comparative results of the algorithms are shown in Table 1.

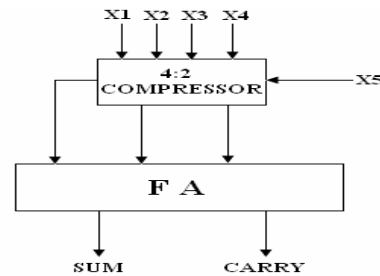


Figure 1. Proposed Approach of Performing Five-to-two CSA Operation

3.2 Improvement of Algorithm 2 Proposed in [5]

The second algorithm proposed in [5] is based on the use of a four-to-two rather than a five-to-two CSA, thus saving a full level of CSL. The critical delay of this algorithm is based on the two-level of CSL. In the proposed approach the computation is reduced to one level through the use of novel hardware unit. The proposed hardware unit can perform the four-to-two CSA operation singly. The architecture of the proposed novel hardware unit is shown in Figure 2. In the proposed hardware unit, the carry save representation of the four input operands is output in the form of SUM and Carry in 3XOR delays. Thus the critical delay of the algorithm is

$$1 \text{ Novel Hardware Unit} + 4:1 \text{ Multiplexer} + 2 \text{ XOR} + 1 \text{ AND} \\ = 3 \text{ XOR} + 2 \text{ XOR} + 4:1 \text{ MUX} + 1 \text{ AND} \\ = 5 \text{ XOR} + 4:1 \text{ MUX} + 1 \text{ AND Gate.}$$

The comparative results of the algorithms are shown in Table 1.

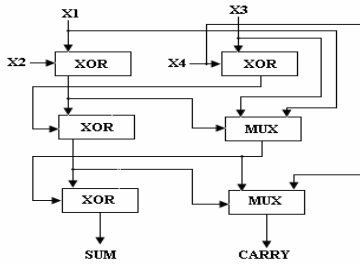


Figure 2. Proposed Novel Hardware unit having propagation delay of 3 XOR gates

4. Optimal Transistor Implementations of the Proposed Improvements

The authors also propose the optimal transistors implementation of the proposed architectures. The authors have recently proposed low power high throughput 4:2 compressor with minimum number of transistors [3]. Furthermore, a low power 12T(transistor) multiplexer based full adder is also proposed recently[4]. The authors have combine the optimal designs of 4:2 compressor and multiplexer based full adder to give the optimal transistor implementation of the proposed approach of performing Five-to-two CSA operations. Figure 3 shows the optimal transistor implementation of the proposed approach of performing Five-to-two CSA operations. The authors have also proposed the optimal transistor implementation of the proposed novel hardware unit performing four-to-two CSA operations. Figure 4 shows the optimal transistor implementation of the proposed novel hardware unit used to perform four-to-two CSA operations. In the transistor implementation of the proposed novel hardware unit, the four transistor XOR gate proposed in [6] is used for performing XOR operations. The proposed transistor implementations are highly optimized in terms of area, speed and low power.

5. Discussions and Conclusions

The proposed approaches improve the Algorithm 1 and Algorithm 2 proposed in [5] significantly. Table 1 clearly signifies the dominance of the proposed approaches compared to existing one in [5]. As shown in Table 1, there is saving 1 XOR gate delay, when proposed approaches are used for a single modular multiplication. Hence, they will be of eminent value when implemented for higher word length such as 1024 and 2048, as there will be saving in the propagation delays by 1024 and 2048 XOR gates respectively. The designs presented are

technology independent and thus can be ported to other silicon technologies without difficulty.

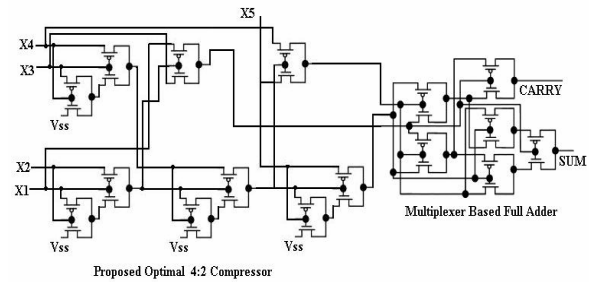


Figure 3. Transistor Implementation of Proposed Approach of Performing Five-to-two CSA Operation

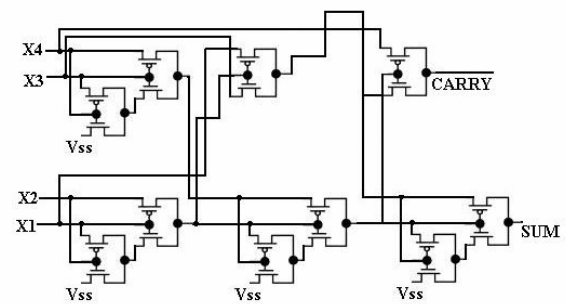


Figure 4. Transistor Implementation of Proposed novel hardware unit

6. References

- [1]Montgomery, P.L.: ‘Modular Multiplication without Trial Division’, *Math. Comput.*, 1985, 44, pp. 519–521.
- [2] V. Oklobdzija, "High-Speed VLSI Arithmetic Units: Adders and Multipliers", in "Design of High-Performance Microprocessor Circuits", Book Chapter, Book edited by A. Chandrakasan, IEEE Press, 2000.
- [3] Himanshu Thapliyal, Pallavi Gopineedi and M.B Srinivas, “Novel and efficient 4:2 and 5:2 compressors with minimum number of transistors designed for low-power operations”, SPIE Microelectronics, MEMS, and Nanotechnology Symposium, Brisbane, Australia,11-14 December 2005.(Accepted)
- [4] Yingtao Jiang et.al,"A Novel Multiplexer-Based Low-Power Full Adder", *IEEE Transactions on Circuits and Systems—II: Express Briefs*, vol. 51, no. 7, July 2004 .
- [5]C. McIvor, M. McLoone and J.V. McCanny, “Modified Montgomery modular multiplication and RSA exponentiation techniques”, *IEE Proc.-Comput. Digit. Tech.*, Vol. 151, No. 6, November 2004.
- [6] H. T. Bui, A. K. Al-Sheraidah, and Y. Wang, “Design and analysis of 10-transistor full adders using novel XOR-XNOR gates,” in *Proc. Int. Conf. Signal Processing 2000 (Wold Computer Congress)*, Beijing, China, Aug. 2000.