Performance Comparison of Finite Field Multipliers for SM2 Algorithm based on FPGA Implementation

Munkhbaatar Chinbat[1,2], Liji Wu[1,2*], Altantsooj Batsukh[1,2], Uyangaa Khuchit[1,2], Xiangmin Zhang[1,2], Bayarpurev Mongolyn[3], Ke Xu[4], Wei Yang[4,]

[1] Beijing National Research Center for Information Science and Technology
[2] Institute of Microelectronics, Tsinghua University, Beijing 100084, China
[3] Department of Electronics and Communication Engineering,
National University of Mongolia, Ulaanbaatar 14200, Mongolia
[4] ZTE Corporation, 11F, ZTE R&D Building, No.55 Science and Technology South Street,
Shenzhen, Guangdong, 518057, China
*lijiwu@mail.tsinghua.edu.cn

***Abstract*—An efficient implementation of the multiplication part is one of the significant procedures of the cryptography algorithms. In this paper, the six altered parallel multiplication methods are proposed to implement in 192-bit for the SM2 algorithm. The CPAM, CSAM, Tri-Section Pezaris, Baugh-Wooley array, Modified Booth, and the Montgomery multipliers are compared by considering minimum operational speed, area, and power. We used a *mod m* reducer circuit for comparing with similar outputs of the multiplier architectures. Through the final comparison, the Montgomery gives the efficient result by 504 LUTs, 5.532ns timing, 0.101mW dynamic power. The proposed work is implemented on the Xilinx Virtex-7 FPGA board, and the programming language is VHDL.**

Keywords—ECC; SM2; finite field multipliers; Montgomery

I. Introduction

A public-key encryption (PKE) system uses two keys, public and private ones. The encryption part works by taking a message and applying a mathematical operation to get ciphertext. The decryption part takes the encrypted text and involves a different operation to recover the original message. The public key is used for encryption, and the private key is used for decryption. Victor Miller and Neal Koblitz proposed elliptic curve cryptography (ECC) in 1985. ECC is based on computational operations of elliptic curves and uses the position of points on an elliptic curve to encrypt and decrypt information[1]. ECC with small key sizes for higher security compared with RSA, which makes them suitable for digital signature and key agreement algorithm ECC with smaller key sizes, has proven its security rather than RSA, which marks it suitable for digital signature and key agreement algorithm[2].

SM2 is a public-key cryptographic algorithm, which is based on ECC. The State Cryptography Administration announces it expands the international standard ECC algorithm in 2010[3].

The most operational calculations on SM2 are multipliers and adders. Particularly the chosen multiplier design could define time-consuming and the design array values. In this paper, altered types of parallel array multipliers equalize with the Montgomery multiplication algorithm, commonly used for the PKE system. The chosen parallel array multipliers are the Carry Propagate, the Carry Save, the Tri-Section Pezaris, the Baugh Wooley, and Modified Booth array multiplier to accomplish with the reducer module for the multiplications on a finite field.

We implemented selected methods for SM2 algorithm with 192-bit multiplication, and the key parameters are compared on 40nm Xilinx Virtex-7 FPGA by the main parameters such as power, timing, and area.

This paper is organized as follows. Section II introduces significant notations and algorithms about the multiplier. Section 0 reveals the proposed multiplication algorithms for the experiments. Implementation setting, results, and comparisons of this work is explained in Section IV. Finally, the conclusion is given in Section V.

II. Preliminaries

A. Galois Field

A finite number of elements compose Galois Field (GF) or known as finite field. Representing data as a vector in a GF allows several operations such as add, multiply, and inverse operations. GF(2) is a binary field, and it can be extended to GF(2^k), where the elements are 0 and 1. These two fields are most widely used in digital data transmission and storage system[4]. The multiplication is more complicated rather than add operation since the multiplication of two polynomial *g(x)* and *f(x)*. Galois multiplier equation is given as “(1)”.

$$m(x) = (g(x) * f(x)) \bmod(p(x)) \qquad (1)$$

The product of multiplication is the reminder of the multiplication result *f(x)*g(x)* divided by the primitive polynomial of the field. The ECC algorithm’s performance depends on the arithmetic in the underlying GF.

978-1-7281-6879-1/20/$31.00 ©2020 IEEE

B. Multiplier

Multiplication is a basic arithmetic operation whose execution is composed of multiplier and multiplicand. Most cryptographic algorithms include the multiplication process in the architectures, as well as for fulfilling fast and cost-effective multiplier design that is critical on hardware implementation. FPGA implementations of arithmetic functions such as multiplier is presented in[5]. In this work, we used parallel multiplier methods for comparing on FPGA, and the general algorithm of parallel multiplication is shown in Algorithm 1. In this algorithm, *a, b* are integers. If representing *a* by a whole integer number, it is becoming a=$2^{m-1} A[m-1] + ... + 2^2 A[2] + 2 A[1] + A[0]$.

Algorithm 1. Multiplier Algorithm

```
Input $a, b \in [0, p-1]$.
Output $c = a \cdot b$
1.  $Set\ C[i] \leftarrow 0\ for\ 0 \le i \le t \le t-1$.
2.  For $i$ from 0 to $t$-1 do
    2.1. $U \leftarrow 0$.
    2.2. $For\ j\ from\ 0\ to\ t-1\ do$:
           $(UV) \leftarrow C[i+j] + A[i] \cdot B[j] + U$.
           $C[i+j] \leftarrow V$.
    2.3. $C[i+j] \leftarrow U$.
3.  Return $(c)$.
```

Altered multiplier algorithm architectures are presented in Section III.

III. Architecture of Multiplication

In this paper, we have chosen to compare six different multiplication methods, built by 192*192-bit for SM2 algorithm. Subsequently, the 384-bit product would modify into 192-bit by *mod m* reducer, except Montgomery multiplication. Various multiplications and reducer architectures are shown in the following figures, respectively.

A. Mod m Reducer

For fixed values of *m*, specific combinational mod *m* reducers can be considered. We used *mod m* multiplier, where $m = 2^{192} - 2^{64} - 1$. Any 192*192-bit multipliers can use this mod *m* reducer circuit. The reducer circuit is shown in Fig. 1.

B. Parallel Multiplication Methods

Parallel multiplication produces the simultaneous generation of all partial products, which have to be added together to get the result. The building blocks of the parallel multiplier is the AND gate, the Full Adder (FA) and the Half Adder (HA)[6], respectively.

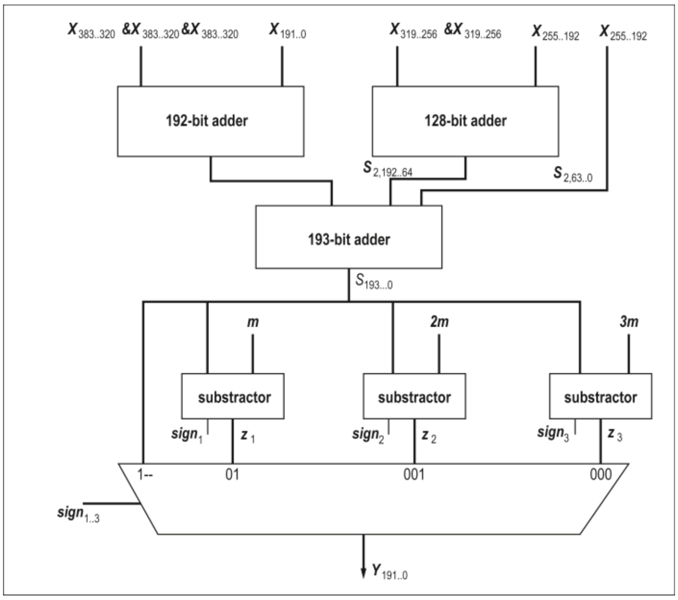

Figure 1. mod 2^{192} - 2^{64} – 1 reducer

a) Carry Propagate Array Multiplier (CPAM)

CPAM is the parallel multiplier for unsigned operands multiplier that uses the carry-propagate adders to make the required additions. In CPAM, the AND gates are used to derive the partial product bits, whereas the FAs and HAs are used for the addition of the partial products[6]. The architecture of the 192*192-bit CPAM circuit is presented in Fig. 2(a), in which *x* and *y* are the input bits of the multiplicands and multipliers, and the input line $pp_{i,j}$ denotes the output of an AND gate.

b) Carry Save Array Multiplier (CSAM)

It is composed of 2-input AND gates as same as CPAM, by using an array of carry-save adders for adding them and a ripple-carry adder for producing the final product. The carry save-adder is one of the ways for acceleration the multiplication. In Fig. 2(b), the architecture of 192 by 192-bit CSAM circuit is shown.

c) Tri-Section Pezaris Array Multiplier (TPAM)

Tri-Section Pezaris Multiplier is referred in "direct two's complement array multiplication" method. It uses different types of FAs in the implementation circuits. Without the complementing stages, the timing of the multiplication process is decreased, and power consumption is increased. Tri-Section Pezaris Multiplier is signed number multiplication, which is shown in Fig. 2(c).

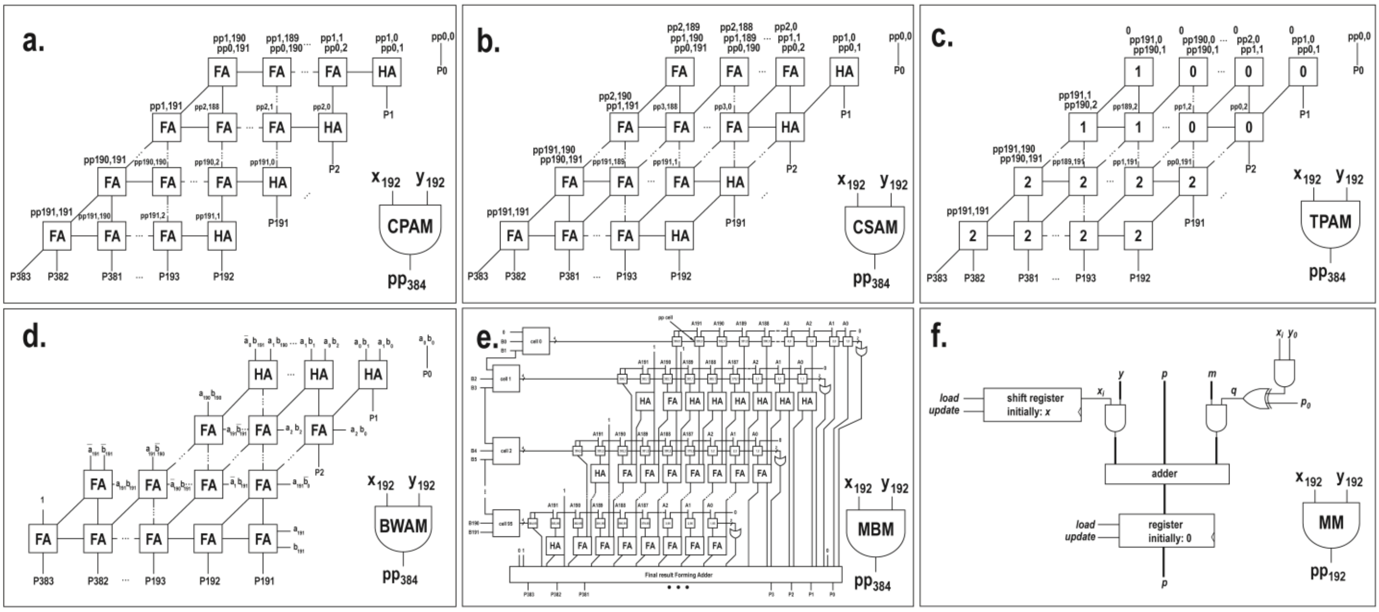

Figure 2. The architectures of multipliers: (a).CPAM, (b) CSAM, (c) TPAM, (d) BWAM, (e) MBM and (f) MM

d) Baugh-Wooley Array Multiplier (BWAM)

Baugh-Wooley is similar to Tri-Section Pezaris multiplier. These two methods are mostly used in the application where consists of small bits of operands. The advantage of the direct two's complement multiplication algorithms is the more straightforward structure of the architecture rather than other parallel multiplication methods such as the Booth multiplier. Furthermore, the signs of all additions are positive, which allows the fluent structure for hardware implementation. The architecture of 192*192-bit BWAM is presented in Fig. 2(d).

e) Modified Booth Multiplier (MBM)

Modified booth multiplier is a parallel multiplier for two's complement notation. The two's complementation logic is based on the idea of the Sklansky parallel prefix tree[7], which can finish two's complementation of binary number in a logarithmic time O(Log$_2$n). MBM uses three main steps, which are 2-bit Booth encoding, carry-save adders for adding the partial products, and a ripple carry adder for producing the final product. Fig. 2(e) shows the MBM architecture.

C. Montgomery Multiplier (MM)

This multiplication is described in[8]. The advantage of this algorithm is to division with shift and addition. The Montgomery multiplication architecture is illustrated in Fig. 2(f). Peter Montgomery proposed this algorithm, which is improved from the classic parallel algorithms, where the carry propagation is gone from the key bits, redundancy is avoided and simultaneous broadcasting of digits is no longer required[9]. These remarkable modifications are able to reduce the area, and also the critical path in the implementation.

IV. Implementation

Our design was implemented on the Xilinx Virtex-7 family xc7vx1140tflg1930-1 FPGA as the hardware device.

The experiment is completed on the Vivado 2018.1 design tool in VHDL to synthesize our design. The implementation and performance comparison of 192-bit multiplier methods are presented in TABLE I.

As mentioned in Section 0, from 1 to 5 numberings in TABLE I. that are classic parallel array multiplications and 6 is an improved algorithm which is widely used for public-key cryptography, nowadays. For implementing CPAM, CSAM, TPAM, BWAM, and MBM, we used *mod* 2^{192}- 2^{64}– *1* reducer circuit to reduce the output bits of the products. Montgomery multiplication has a reducer circuit on itself. As shown in TABLE I. and Fig. 3, the comparisons of operating speeds, area, and power consumption are presented, respectively.

The achieved result of CSAM shows that higher LUTs, power consumption, and lower timing values than the CPAM, the TPAM, and BWAM, whereas the CSAM architecture is based on both carry-save adder and ripple-carry adder. The TPAM and BWAM algorithms are suitable for lower bits of operands such as an 8-bit application. The MBM includes the 2-bit booth encoding scheme that allows considerable performance rather than previous four algorithms.

TABLE I. IMPLEMENTATION RESULTS OF THE PROPOSED MULTIPLICATIONS

	Methods	LUTs	Timing (ns)	Dynamic power (mW)
1	CPAM	69594	322.436	3.967
2	CSAM	88579	139.046	5.778
3	TPAM	81945	297.644	4.883
4	BWAM	73284	170.725	4.406
5	MBM	43072	136.493	3.059
6	MM	504	5.532	0.101

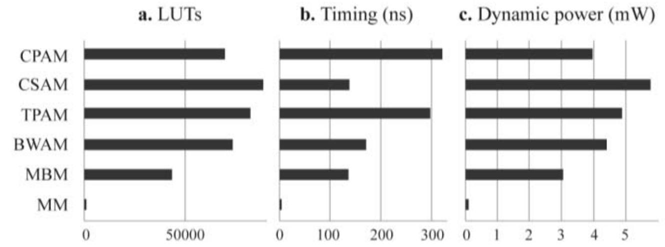

Figure 3. Comparisons by main parameters

Furthermore, die areas of the proposed algorithms are shown in Fig. 4. The listed method numbers in the TABLE I. is the same as those in Fig. 4. However, from 1 to 5 algorithms used the separated reducer circuit, the Montgomery design scales area down to many times rather than the first five parallel multiplications. Based on the main parameter results, the Montgomery multiplication is verified as an efficient method for SM2 algorithm.

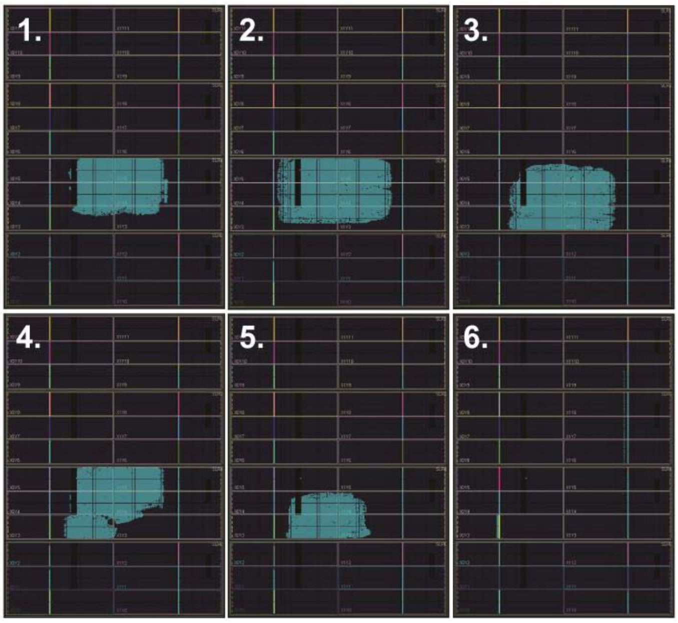

Figure 4. Die areas of proposed multiplications

V. Conclusion

Multiplication is a crucial part of the cryptography system. Six altered multiplier methods are presented for SM2 algorithm, where each method is implemented in 192-bit on Xilinx Virtex-7 FPGA.

We adapted CPAM, CSAM, TPAM, WBAM, and MBM methods into 192-bit architecture multipliers, which were enhanced by the mod m reducer module for the SM2 algorithm. As the results of power consumption, die area size, and timing speed of all multipliers are revealed, the Montgomery multiplier is superior performance than parallel array multipliers in the PKE system.

References

[1] Y. Yin, L. Wu, Q. Peng, and X. Zhang, "A Novel SPA on ECC with Modular Subtraction," in *2018 12th IEEE International Conference on Anti-counterfeiting, Security, and Identification (ASID)*, pp. 179–182, 2018.

[2] Dr. S. Arivazhagan, Mrs. W. Sylvia Lilly Jebarani, Ms. S. Veera Kalyani, and Ms. A. Deiva Abinaya, "Locally Applied Mixed Chaotic Maps based Encryption for High Crypto Secrecy," *International Journal of Engineering Research and*, vol. V6, no. 04, Apr. 2017.

[3] D. Yang, Z. Dai, W. Li, and T. Chen, "An Efficient ASIC Implementation of Public Key Cryptography Algorithm SM2 Based on Module Arithmetic Logic Unit," in *2019 IEEE 13th International Conference on ASIC (ASICON)*, pp. 1–4, 2019.

[4] S. A. Mozhi and P. Ramya, "Efficient bit-parallel systolic multiplier over GF (2m)," Int. Conf. Electr. Electron. Optim. Tech. ICEEOT 2016, pp. 4800–4803, 2016

[5] J.-P. Deschamps, G. D. Sutter, and E. Cantó, *Guide to FPGA Implementation of Arithmetic Functions*, vol. 149. Dordrecht: Springer Netherlands, 2012.

[6] D. Bakalist, X. Kavousianos, H. T. Vergos, D. Nikolos, and G. P. Alexiou, "Low Power Built-In Self-Test Schemes for Array and Booth Multipliers," *VLSI Design*, vol. 12, no. 3, pp. 431–448, Jan. 2001.

[7] A. Prasath A.M., R. V. Arjun, K. Deepaknath, and K. Gayathree, "Implementation of optimized digital filter using sklansky adder and kogge stone adder," in *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pp. 661–664, 2020.

[8] P. L. Montgomery, "Modular multiplication without trial division," *Mathematics of Computation*, vol. 44, no. 170, pp. 519–519, May 1985.

[9] C. D. Walter, "Montgomery's Multiplication Technique: How to Make It Smaller and Faster," pp. 80–93, 1999.