International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 12, Issue 02, February 2023)

VLSI Architecture of Montgomery Modular Multiplier for FPGA Application

Pranshu Jaiswal[1], Dr. Laxminarayan Gahalod[2]

[1]*Research Scholar,* [2]*Associate Professor, Department of Electronics and Communication Engineering, Lakshmi Narain College of Technology, Bhopal, India*

***Abstract*—Multiplier plays a key operation to check the performance of any processor. This work proposes a simple and efficient Montgomery multiplication algorithm such that the low-cost and high-performance Montgomery modular multiplier can be implemented accordingly. The proposed Montgomery modular multiplier receives and outputs the data with binary representation and uses only one-level carry-save adder (CSA) to avoid the carry propagation at each addition operation. Experimental results show that the proposed Montgomery modular multiplier can achieve higher performance and significant area time product improvement when compared with previous designs.**

***Keywords*—Montgomery, Modular, VLSI, Multiplier.**

I. INTRODUCTION

Montgomery modular multiplication, all the more generally alluded to as Montgomery multiplication, is a technique for performing quick modular multiplication. Given two numbers an and b and modulus N, the established modular multiplication calculation registers the twofold width item stomach muscle mod N, and after that plays out a division, subtracting products of N to offset the undesirable high bits until the rest of by and by not as much as N. Montgomery decrease rather adds products of N to offset the low bits until the outcome is a various of a helpful (for example power of two) consistent R > N. At that point the low bits are disposed of, creating an outcome under 2N. One last restrictive subtract decreases this to not as much as N.

The outcome is the ideal item separated by R, which is less badly designed than it may show up. To increase an and b, they are first changed over to Montgomery structure or Montgomery portrayal aR mod N and bR mod N. Whenever increased, these produce abR2 mod N, and the accompanying Montgomery decrease produces abR mod N, the Montgomery type of the ideal item. Changing over to and from Montgomery structure makes this slower than the ordinary or Barrett decrease calculations for a solitary duplicate.

Be that as it may, when performing numerous multiplications consecutively, as in modular exponentiation, middle of the road results can be left in Montgomery structure, and the underlying and last transformations turn into an irrelevant portion of the general calculation. Numerous essential cryptosystems, for example, RSA and Diffie– Hellman key trade depend on math activities modulo an extensive number, and for these cryptosystems, the calculation by Montgomery multiplication is quicker than the accessible alternatives.

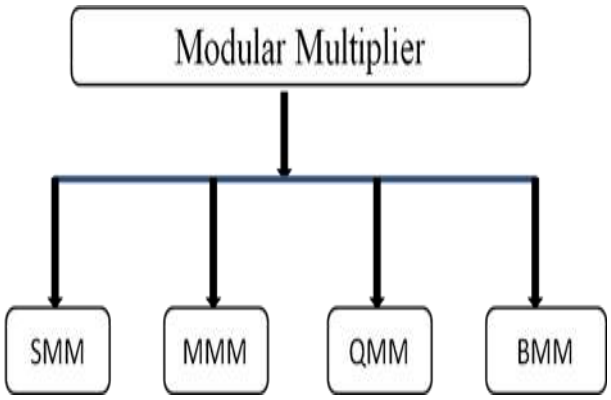

Figure 1: Classification of modular multiplier

In figure 1, showing different types of modular multiplier. Systolic Modular Multiplication (SMM), Montgomery modular multiplier (MMM), Quantum Modular Multipliers (QMM), Barrett Modular Multiplier (MMMM)

Multiplication: Cryptographic applications don't use negative numbers; therefore our digit-multiplication circuit performs only unsigned multiplications. The products are accumulated (added to a 32…50-bit register) but only single digits are extracted from these registers and stored in memory.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 12, Issue 02, February 2023)

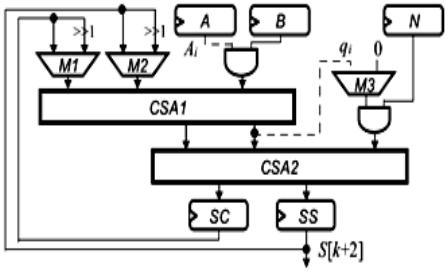

Figure 2: Low cost High performance VLSI architecture

For operand sizes in cryptographic applications the school multiplication is the best, requiring simple control. Some speed improvement can be expected from the more complicated Karatsuba method, but the Toom-Cook 3-way (or beyond) multiplication is actually slower for these lengths. An FFT based multiplication takes even longer until much larger operands (in our case about 8 times slower).

II. Proposed Methodology

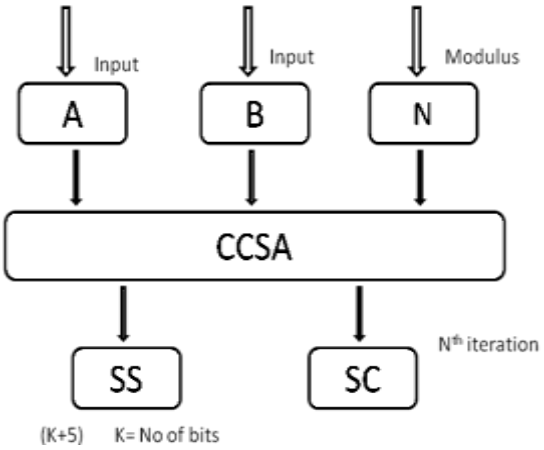

Figure 3: Flow Chart

Steps-

- Proposed MMM consists of one level configurable carry save adder (CCSA) architecture.
- In proposed MMM, there are A and B input bits and N is Modulus bit.
- Apply input bits as well as modulus bit in CCSA architecture.
- Now it process input bits and perform operation.
- Generate save sum (SS) and save carry (SC)
- This process continues till N^{th} number of iteration.
- Sum will save after every iteration and carry will be zero at the k+5 repetitions.
- Finally, SS[k +5] in binary format is outputted
- when SC[k + 5] is equal to 0

In figure 2, A configurable CSA (CCSA), which could be one full-adder or two serial half-adders, is proposed to reduce the extra clock cycles for operand pre computation and format conversion by half, The proposed multiplier used one-level CCSA architecture and skipped the unnecessary carry-save addition operations to largely reduce the critical path delay and required clock cycles for completing one MM operation

It is assume $m = \{m_{n-1}\, m_{n-2} \ldots m_0\}$ is normalized, that is $\frac{1}{2}\, d \le m_{n-1} < d$ or $\frac{1}{2}\, d^{n-1} \le m < d^n$. It is normally the case with RSA moduli. If not, it is have to normalize it: replace m with $2^k m$. A modular reduction step (discussed below) fixes the result: having $R_k = a \mod 2^k m$ calculated, $R \leftarrow R_k - q \cdot m$, where q is computed from the leading digits of R_k and $2^k m$. These de/normalization steps are only performed at the beginning and end of the calculations (in case of an exponentiation chain), so the amortized cost is negligible.

III. Simulation And Result

The simulation studies involve the deterministic RTL circuit as shown in Figure 3 and 4. The proposed MMM implemented with Xilinx 14.7 software using verilog programming language.

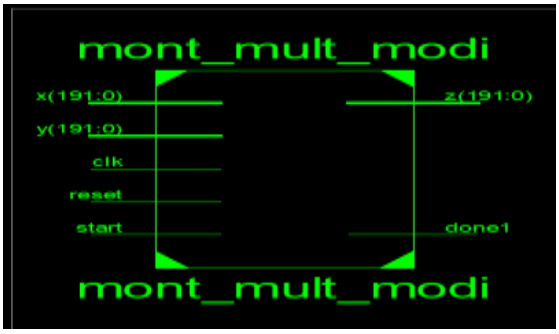

Figure 4: Top module of proposed MMM

Figure 4 present top module of Montgomery Modular Multiplication; it is designed for 192 bits. Other pin configuration is also showing in this diagram.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 12, Issue 02, February 2023)

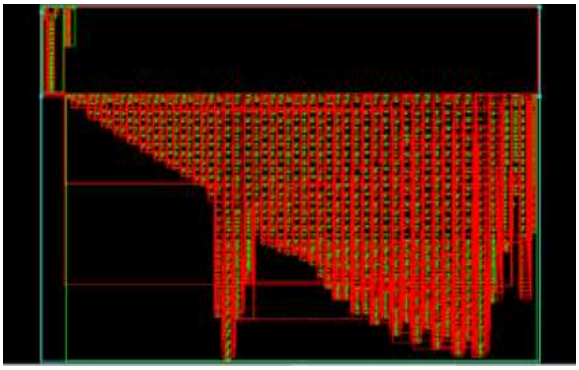

Figure 5: RTL view of proposed MMM

In figure 5 present Register Transfer Level module of Montgomery Modular Multiplication, in which many input output lines and output lines connected with various blocks.

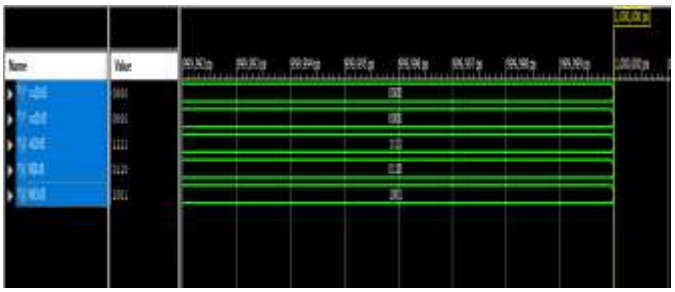

Figure 6: Result validation in test bench for proposed MMM

In figure 6, showing result, on the bases of critical path delay reduction, clock cycle number reduction, and quotient pre computation mentioned above, a new SCS-based Montgomery MM algorithm using one-level CCSA architecture is proposed to significantly reduce the required clock cycles for completing one MM. qi+1 and qi+2 must be generated in the i th iteration, the iterative index i of Montgomery MM will start from −1 instead of 0 and the corresponding initial values of ˆ q and ˆA must be set to 0. Furthermore, the original for loop is replaced with the while loop in SCS-MM-New algorithm to skip some unnecessary iterations when skipi+1 = 1. In addition, the ending number of iterations in SCS-MM-New algorithm is changed to k + 4 instead of k + 1. This is because B is replaced with ˆB and thus three extra iterations for computing division by two are necessary to ensure the correctness of Montgomery MM. In the while loop, The computations of qi+1, qi+2, and skipi+1 in next step and the selections of ˆA , ˆ q, and i in next steps can be carried out in parallel.

The right-shift operations of next steps will be delayed to next clock cycle to reduce the critical path delay of corresponding hardware architecture.

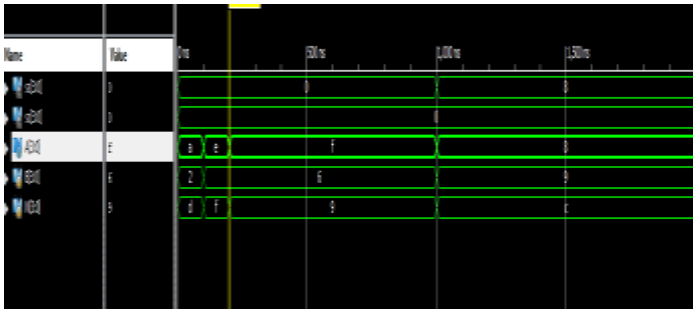

Figure 7: Result validation in different valises for proposed MMM

In figure 7 showing, various values to check output authentication. If SS and SC give 00 in output then it shows that proposed MMM give accurate result. The carry propagation addition operations of B + N and the format conversion are performed by the one-level carry save adder (CSA) architecture of the MSCS-MM multiplier through repeatedly executing the carry-save addition (SS, SC) = SS + SC + 0 until SC = 0

Table 1
Result comparison

Sr No	Parameter	Previous Work	Proposed Work
1	Method	CSA(Carry save adder)	CCSA(Configurable Carry save adder) and Semi Carry Save (SCS)
2	Area	46%	40%
3	Delay	5.60ns	3.768ns
4	Power	0.065mW	0.042mW
5	Time	35 Sec	30.00 Sec

Table 1 is showing comparison of proposed work with previous work, so it can be seen that proposed work gives better result than existing work.

IV. CONCLUSION

This Montgomery Modular multiplier proved to be efficient in the case of area as well as timing constraints. But one more operation of multiplication and modular operation have to be done.

In the parallel operation, for every Montgomery modular multiplier there is additional operation for multiplication and modular operation, which can be avoided by pre-computing R n*M mod p where M is the number of multiplier required and storing that value in a register. This will reduce the clock cycle as well as area in the chip. The pre computation and the format conversion process may lead to additional clock cycles this can increase the critical path, so if the CSA can do a three input addition the additional clock cycles required for the mentioned processes can be made half. Thus the Montgomery multiplier will be having higher efficiency and the hardware takes small area.

REFERENCES

[1] A. A. H. Abd-Elkader, M. Rashdan, E. -S. A. M. Hasaneen and H. F. A. Hamed, "FPGA-Based Optimized Design of Montgomery Modular Multiplier," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 68, no. 6, pp. 2137-2141, June 2021, doi: 10.1109/ TCSII.2020. 3040665.

[2] S. M. -H. Farzam, S. Bayat-Sarmadi, H. Mosanaei-Boorani and A. Alivand, "Fast Supersingular Isogeny Diffie-Hellman and Key Encapsulation Using a Customized Pipelined Montgomery Multiplier," in IEEE Transactions on Circuits and Systems I: Regular Papers, doi: 10.1109/TCSI.2021.3129589.

[3] F. Pajuelo-Holguera, J. M. Granado-Criado and J. A. Gómez-Pulido, "Fast Montgomery Modular Multiplier using FPGAs," in IEEE Embedded Systems Letters, doi: 10.1109/LES.2021.3090029.

[4] M. -H. Farzam, S. Bayat-Sarmadi, H. Mosanaei-Boorani and A. Alivand, "Hardware Architecture for Supersingular Isogeny Diffie-Hellman and Key Encapsulation Using a Fast Montgomery Multiplier," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 68, no. 5, pp. 2042-2050, May 2021, doi: 10.1109/ TCSI.2021. 3062871.

[5] B. Zhang, Z. Cheng and M. Pedram, "High-Radix Design of a Scalable Montgomery Modular Multiplier with Low Latency," in IEEE Transactions on Computers, doi: 10.1109/ TC. 2021.3052999.

[6] J. Ding and S. Li, "A Low-Latency and Low-Cost Montgomery Modular Multiplier Based on NLP Multiplication," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 67, no. 7, pp. 1319-1323, July 2020, doi: 10.1109/ TCSII.2019. 2932328.

[7] G. Gallin and A. Tisserand, "Generation of Finely-Pipelined GF(PP) Multipliers for Flexible Curve Based Cryptography on FPGAs," in IEEE Transactions on Computers, vol. 68, no. 11, pp. 1612-1622, 1 Nov. 2019, doi: 10. 1109/ TC. 2019. 2920352.

[8] Z. Gu and S. Li, "A Division-Free Toom–Cook Multiplication-Based Montgomery Modular Multiplication," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 66, no. 8, pp. 1401-1405, Aug.2019,doi: 10.1109/ TCSII. 2018. 2886962.

[9] S. S. Erdem, T. Yanık and A. Çelebi, "A General Digit-Serial Architecture for Montgomery Modular Multiplication," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 25, no. 5, pp. 1658-1668, May 2017, doi: 10.1109/TVLSI.2017.2652979.

[10] W. Dai, D. D. Chen, R. C. C. Cheung and Ç. K. Koç, "Area-Time Efficient Architecture of FFT-Based Montgomery Multiplication," in IEEE Transactions on Computers, vol. 66, no. 3, pp. 375-388, 1 March 2017, doi: 10.1109/ TC.2016. 2601334.

[11] D. D. Chen, G. X. Yao, R. C. C. Cheung, D. Pao and Ç. K. Koç, "Parameter Space for the Architecture of FFT-Based Montgomery Modular Multiplication," in IEEE Transactions on Computers, vol. 65, no. 1, pp. 147-160, 1 Jan. 2016, doi: 10.1109/TC.2015.2417553.

[12] S. Kuang, K. Wu and R. Lu, "Low-Cost High-Performance VLSI Architecture for Montgomery Modular Multiplication," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 24, no. 2, pp. 434-443, Feb. 2016, doi: 10. 1109/TVLSI.2015.2409113.