Verilog Implementation of Montgomery Modular Multiplication

Narendra Kumar[1], Ambresh Patel[2], Braj Kishor[3]

[1]M.Tech Scholar, [2&3]Assistant Professor
Department of Electronics and Communication,
Sri Satya Sai College of Engineering, RKDF University, Bhopal, India.

Abstract: Montgomery modular multiplication, more commonly referred to as Montgomery multiplication, is a method for performing fast modular multiplication. This work shows a simple and efficient Montgomery multiplication algorithm such that the low-cost and high-performance Montgomery modular multiplier (MMM) can be implemented accordingly. This Paper proposed Verilog Implementation of Montgomery Modular Multiplication. It 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. Simulation results show that the proposed Montgomery modular multiplier can achieve higher performance and significant area time product improvement when compared with existing MMM designs.

Index Terms **- Montgomery, Modular, Multiplication, MMM, VLSI, CSA, Cryptosystem.**

I. INTRODUCTION

Montgomery modular multiplication, even more by and large suggested as Montgomery multiplication, is a methodology for performing snappy modular multiplication. Given two integers an and b and modulus N, the old style modular multiplication computation figures the twofold width thing stomach muscle mod N, and thereafter plays out a division, subtracting results of N to balance the unfortunate high bits until the remainder is before long not as much as N. Montgomery decline instead adds results of N to balance the low bits until the result is a different of a supportive (for instance intensity of two) enduring R > N. By then the low bits are discarded, producing a result under 2N. One final prohibitive subtract diminishes this to not as much as N. This framework maintains a key good ways from the multifaceted idea of remainder digit estimation and amendment found in standard division counts.

The result is the perfect thing isolated by R, which is less inconvenient than it might appear. To copy an and b, they are first changed over to Montgomery structure or Montgomery depiction aR mod N and bR mod N. At whatever point copied, these produce abR2 mod N, and the following Montgomery diminishing produces abR mod N, the Montgomery kind of the perfect thing. Converting to and from Montgomery structure makes this more delayed than the standard or Barrett decline figurings for a single copy. In any case, when performing various multiplications in progression, as in modular exponentiation, intermediate results can be left in Montgomery structure, and the initial and final changes become an insignificant piece of the general estimation. Various critical cryptosystems, for instance, RSA and Diffie–Hellman key exchange rely upon calculating undertakings modulo an enormous number, and for these cryptosystems, the count by Montgomery multiplication is speedier than the open decisions.

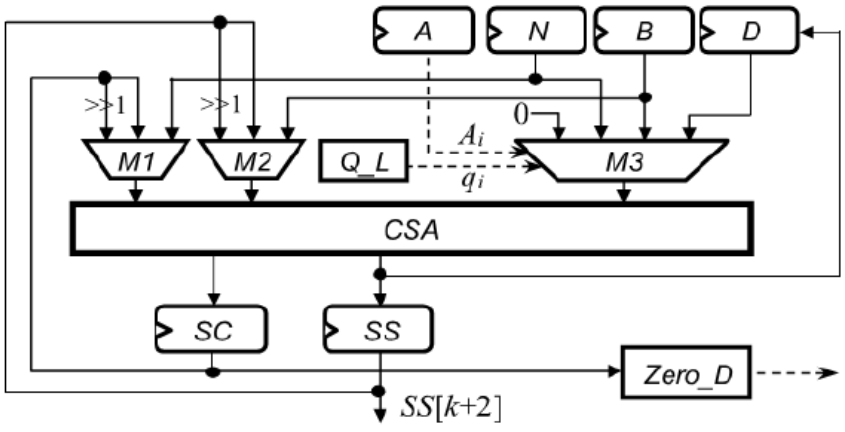

Figure 1: SCS based Montgomery multiplier

For operand measures in cryptographic applications the school multiplication is the best, requiring essential control. Some speed improvement can be ordinary from the more tangled Karatsuba method, yet the Toom-Cook 3-way (or past) multiplication is very for these lengths. A FFT based multiplication takes impressively longer until significantly greater operands (for this circumstance around numerous occasions more slow).

II. PROBLEM FORMULATION

After reviewing literature following problem is observed-

•Leading hardware cost
•Critical path delay
•Extra clock cycles for completing one modular multiplication
•Montgomery modular multiplier by using normal adder

III. PROPOSED METHODOLOGY

Proposed Montgomery modular multiplier can accomplish higher execution and critical region time item improvement when contrasted and existing designs.

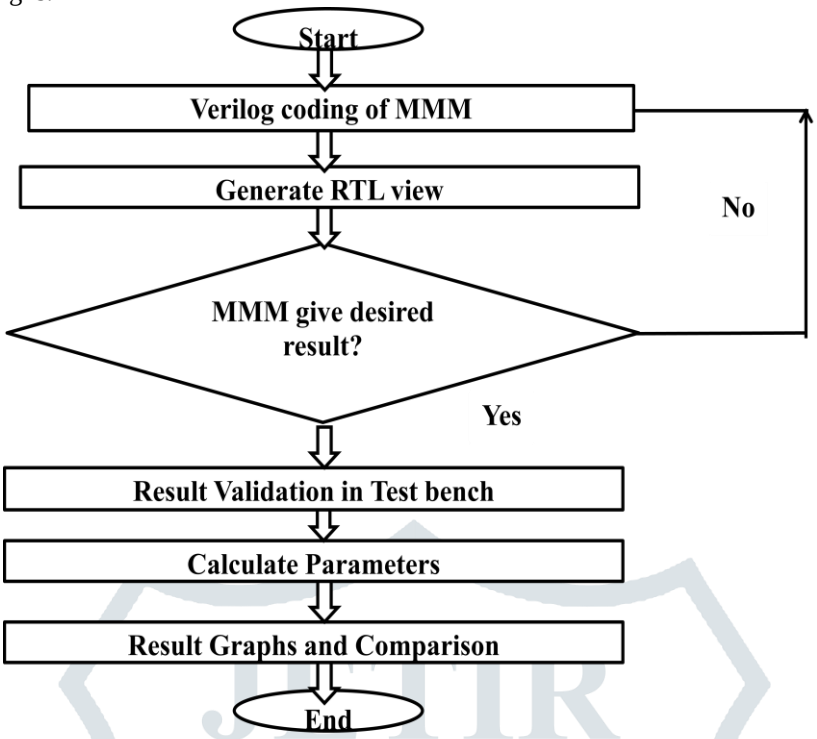

Figure 2: Flow Chart

Steps-

- Proposed MMM comprises of one level configurable carry save adder (CCSA) architecture.
- In proposed MMM, there are An and B input bits and N is Modulus bit.
- Apply input bits just as modulus bit in CCSA architecture.
- Now it procedure input bits and perform activity.
- Generate save sum (SS) and save carry (SC)
- This procedure continues till N^{th} number of cycle.
- Sum will spare after each emphasis and convey will be zero at the k+5 reiterations.
- Finally, SS[k +5] in binary configuration is yielded
- when SC[k + 5] is equivalent to 0

A configurable CSA (CCSA), which could be one full-viper or two sequential half-adders, is proposed to diminish the additional clock cycles for operand pre calculation and organization change significantly, The proposed multiplier utilized one-level CCSA architecture and avoided the superfluous convey spare option activities to generally decrease the basic way deferral and required clock cycles for completing one MM activity

It is assume $m = \{m_{n-1}\, m_{n-2} \ldots m_0\}$ is normalized, that is $\tfrac{1}{2}\, d \le m_{n-1} < d$ or $\tfrac{1}{2}\, d^{n-1} \le m < d^n$. 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 $R_k = a \bmod 2^k m$ 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.

IV. SIMULATION RESULT

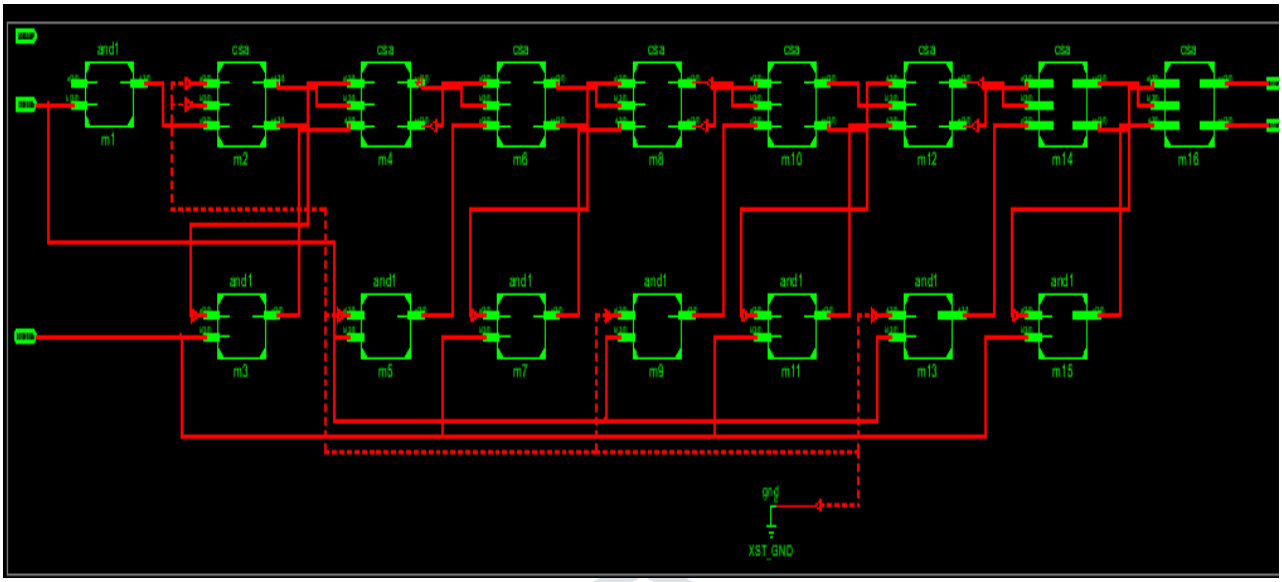

Figure 3: RTL view of proposed MMM

In figure 3, present Register Transfer Level module of Montgomery Modular Multiplication, in which many input output lines and output lines connected with various blocks.

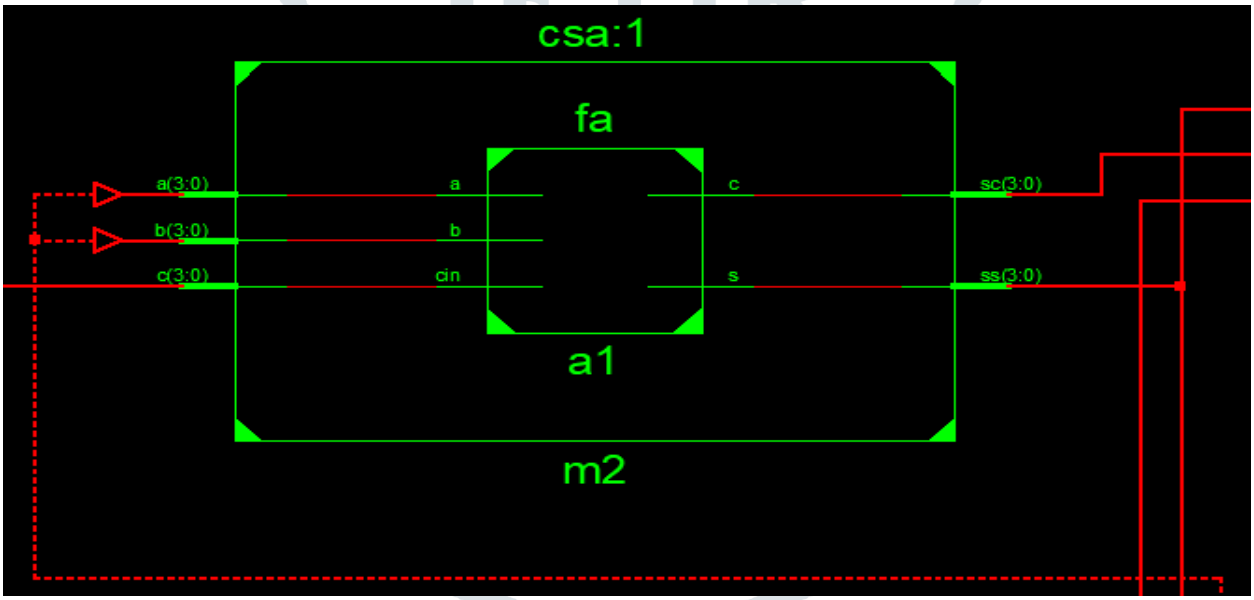

Figure 4: CSA block view of proposed MMM

Figure 4 presents carry save adder of proposed MMM model, here all the operation is based on sum and carry concepts.

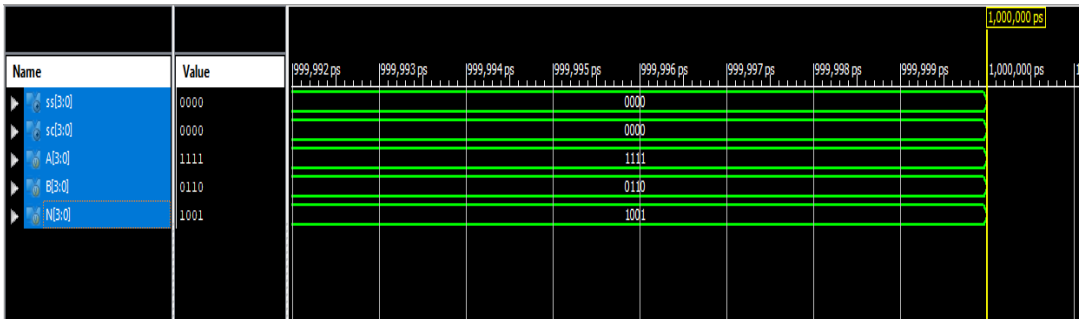

Figure 5: Result validation in test bench for proposed MMM

In figure 5, showing result, on the bases of basic way defer decrease, clock cycle number decrease, and remainder pre calculation referenced over, another SCS-based Montgomery MM calculation using one-level CCSA architecture is proposed to altogether diminish the required clock cycles for completing one MM. qi+1 and qi+2 must be produced in the I the emphasis, the iterative index I of Montgomery MM will begin from −1 instead of 0 and the corresponding initial estimations of ˆ q and ˆA must be set to 0. Moreover, the original for circle is supplanted with the while circle in SCS-MM-New calculation to avoid some pointless

cycles when skipi+1 = 1. Moreover, the ending number of cycles in SCS-MM-New calculation is changed to k + 4 instead of k + 1. This is on the grounds that B is supplanted with ˆB and in this manner three additional cycles for computing division by two are important to guarantee the accuracy of Montgomery MM. In the while circle, The calculations of qi+1, qi+2, and skipi+1 in following stage and the choices of ˆA , ˆ q, and I in subsequent stages can be done in parallel. The right-move activities of following stages will be postponed to next clock cycle to diminish the basic way deferral of corresponding hardware architecture.

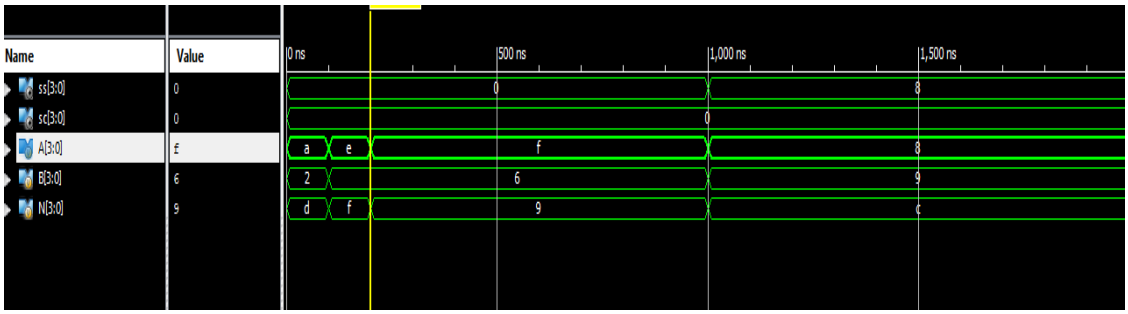

Figure 6: Result validation in different valises for proposed MMM

In figure 6 showing, different qualities to check yield validation. On the off chance that SS and SC give 00 in yield, at that point it demonstrates that proposed MMM give precise outcome. The convey spread expansion activities of B + N and the organization change are performed by the one-level carry save adder (CSA) architecture of the MSCS-MM multiplier through over and again executing the convey spare expansion (SS, SC) = SS + SC + 0 until SC = 0

Table I: Simulation Parameter and Comparison with previous work

Sr No.	Parameter	Previous Work	Proposed Work
1	Method	CSA(Carry save adder)	CSA(Carry save adder) and Semi Carry Save (SCS)
2	Area	46%	42%
3	Delay	5.60ns	3.878ns
4	Power	0.065mW	0.042mW
5	Time	35 Sec	29.00 Sec
6	Memory	5173124 kilobytes	4674588 kilobytes

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.

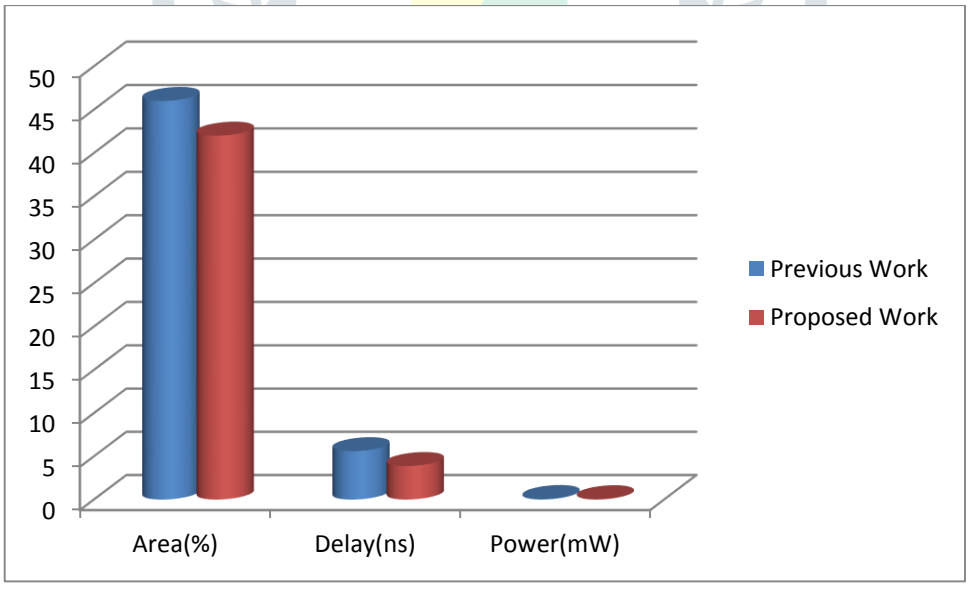

Figure 7: Comparison of previous and proposed work

Figure 7 is showing comparison of area, power and delay of proposed work with existing work.

V. CONCLUSION

Montgomery Modular multiplier demonstrated to be effective for the situation of zone just as timing constraints. In any case, one more activity of multiplication and modular activity must be finished. In the parallel activity, for each Montgomery modular multiplier there is extra activity for multiplication and modular activity, which can be maintained a strategic distance from by pre-computing R n*M mod p where M is the quantity of multiplier required and storing that incentive in a register. This will lessen

the clock cycle just as territory in the chip. The pre calculation and the organization transformation procedure may prompt extra clock cycles this can increase the basic way, so if the CSA can do a three input option the extra clock cycles required for the referenced procedures can be made half. Subsequently the Montgomery multiplier will have higher effectiveness and the equipment takes small area.

REFERENCES

1. 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.
2. 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.
3. J. C. Néto, A. F. Tenca and W. V. Ruggiero, "A Parallel and Uniform k -Partition Method for Montgomery Multiplication," in *IEEE Transactions on Computers*, vol. 63, no. 9, pp. 2122-2133, Sept. 2014.
4. W. Lin, J. Ye and M. Shieh, "Scalable Montgomery Modular Multiplication Architecture with Low-Latency and Low-Memory Bandwidth Requirement," in *IEEE Transactions on Computers*, vol. 63, no. 2, pp. 475-483, Feb. 2014.
5. S. Akleylek, M. Cenk and F. Özbudak, "On the generalisation of special moduli for faster interleaved montgomery modular multiplication," in *IET Information Security*, vol. 7, no. 3, pp. 165-171, Sept. 2013.
6. S. Kuang, J. Wang, K. Chang and H. Hsu, "Energy-Efficient High-Throughput Montgomery Modular Multipliers for RSA Cryptosystems," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 21, no. 11, pp. 1999-2009, Nov. 2013.
7. J. Ye, T. Hung and M. Shieh, "Energy-efficient architecture for word-based Montgomery modular multiplication algorithm," *2013 International Symposium onVLSI Design, Automation, and Test (VLSI-DAT)*, Hsinchu, 2013, pp. 1-4.
8. M. Huang, K. Gaj and T. El-Ghazawi, "New Hardware Architectures for Montgomery Modular Multiplication Algorithm," in *IEEE Transactions on Computers*, vol. 60, no. 7, pp. 923-936, July 2011.
9. Z. Chen and P. Schaumont, "A Parallel Implementation of Montgomery Multiplication on Multicore Systems: Algorithm, Analysis, and Prototype," in *IEEE Transactions on Computers*, vol. 60, no. 12, pp. 1692-1703, Dec. 2011.
10. A. Ibrahim, F. Gebali, H. Elsimary and A. Nassar, "Processor Array Architectures for Scalable Radix 4 Montgomery Modular Multiplication Algorithm," in *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 7, pp. 1142-1149, July 2011.
11. G. D. Sutter, J. Deschamps and J. L. Imana, "Modular Multiplication and Exponentiation Architectures for Fast RSA Cryptosystem Based on Digit Serial Computation," in *IEEE Transactions on Industrial Electronics*, vol. 58, no. 7, pp. 3101-3109, July 2011.
12. M. Shieh and W. Lin, "Word-Based Montgomery Modular Multiplication Algorithm for Low-Latency Scalable Architectures," in *IEEE Transactions on Computers*, vol. 59, no. 8, pp. 1145-1151, Aug. 2010
13. A. Ibrahim, F. Gebali, H. El-Simary and A. Nassar, "High-performance, low-power architecture for scalable radix 2 montgomery modular multiplication algorithm," in *Canadian Journal of Electrical and Computer Engineering*, vol. 34, no. 4, pp. 152-157, Fall 2009.
14. M. Kaihara and N. Takagi, "Bipartite Modular Multiplication Method," in *IEEE Transactions on Computers*, vol. 57, no. 2, pp. 157-164, Feb. 2008.
15. O. Nibouche, A. Bouridane and M. Nibouche, "Architectures for Montgomery's multiplication," in *IEE Proceedings - Computers and Digital Techniques*, vol. 150, no. 6, pp. 361-368, 17 Nov. 2003.