2nd International Conference on
Contemporary Technological Solutions towards fulfillment of Social Needs

Review of Montgomery Modular Multiplication in VLSI Architecture

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

[1]M.Tech Scholar, [2&3]Assistant Professor & Electronics and Communication Department & RKDF University, India

Abstract—This paper presents an overview on execution of Montgomery modular multiplication calculation using VLSI architecture. The Montgomery calculation is a quick modular multiplication technique as often as possible utilized in cryptographic applications, in which the proficiency of cryptosystem relies upon the speed of modular multiplication. This review gives the examination between various alterations done in Montgomery modular multiplication.

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

I. INTRODUCTION

Montgomery modular multiplication, all the more ordinarily alluded to as Montgomery multiplication, is a technique for performing quick modular multiplication. Given two integers an and b and modulus N, the old style modular multiplication calculation registers the twofold width item stomach muscle mod N, and afterward plays out a division, subtracting products of N to offset the undesirable high bits until the remainder is by and by not as much as N. Montgomery decrease instead adds products of N to offset the low bits until the outcome is a various of a helpful (for example intensity of two) consistent R > N. At that point the low bits are disposed of, producing an outcome under 2N. One final contingent subtract diminishes this to not as much as N. This method maintains a strategic distance from the intricacy of remainder digit estimation and remedy found in standard division calculations.

The outcome is the ideal item partitioned by R, which is less inconvenient than it may show up. To duplicate an and b, they are first changed over to Montgomery structure or Montgomery portrayal aR mod N and bR mod N. Whenever duplicated, these produce abR2 mod N, and the following Montgomery decrease produces abR mod N, the Montgomery type of the ideal item. Converting to and from Montgomery structure makes this slower than the regular or Barrett decrease calculations for a single duplicate. In any case, when performing numerous multiplications in succession, as in modular exponentiation, intermediate outcomes can be left in Montgomery structure, and the initial and final changes become an insignificant division of the general calculation. Numerous significant cryptosystems, for example, RSA and Diffie–Hellman key trade depend on math tasks modulo an enormous number, and for these cryptosystems, the calculation by Montgomery multiplication is quicker than the accessible choices.

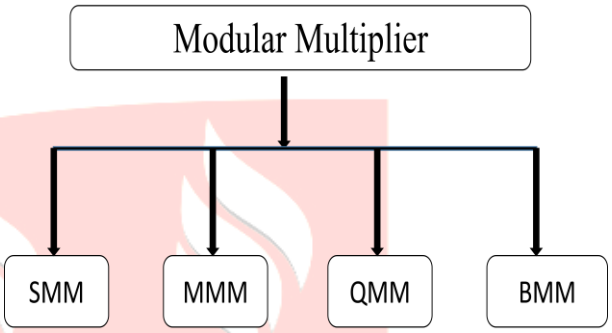

Figure 1: Classification of modular multiplier

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 utilize negative numbers; consequently our digit-multiplication circuit performs just unsigned multiplications. The items are aggregated (added to a 32… 50-bit register) yet just single digits are separated from these registers and put away in memory.

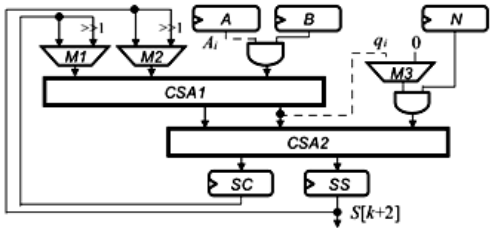

Figure 2: Low cost High performance VLSI architecture

For operand measures in cryptographic applications the school multiplication is the best, requiring basic control. Some speed improvement can be normal from the more entangled Karatsuba technique, however the Toom-Cook 3-way (or past) multiplication is entirely for these lengths. A FFT based multiplication takes significantly longer until a lot bigger operands (for our situation around multiple times slower).

II. LITERATURE SURVEY

S. S. Erdem et al., [1] The Montgomery calculation is a quick modular multiplication technique every now and again utilized in cryptographic applications. This paper investigates the digit-sequential executions of the Montgomery calculation for huge integers. A nitty gritty examination is given and a tight upper headed is exhibited for the intermediate outcomes obtained during the digit-sequential calculation. In light of this investigation, a productive digit-sequential Montgomery

modular multiplier architecture using convey spare adders is proposed and its multifaceted nature is exhibited. In this architecture, pipelined convey select adders are utilized to perform two final errands: adding convey spare vectors representing the modular item and subtracting the modulus from this option, if further decrease is required. The proposed architecture can be intended for any digit size δ and modulus θ. This paper likewise introduces rationale equations for the bits of the precomputation - θ - 1 mod 2 δ utilized in the Montgomery calculation for $\delta \leqslant 8$. Finally, assessment of the proposed architecture on Virtex 7 FPGAs is displayed.

S. Kuang et al.,[2]This paper proposes a straightforward and effective Montgomery multiplication calculation with the end goal that the ease and elite Montgomery modular multiplier can be executed accordingly. The proposed multiplier gets and yields the information with binary portrayal and uses just one-level convey spare viper (CSA) to keep away from the convey spread at every expansion task. This CSA is additionally used to perform operand precomputation and arrangement change from the convey spare organization to the binary portrayal, leading to a low equipment cost and short basic way delay to the detriment of additional clock cycles for completing one modular multiplication.

E. C. Culau et al., [3] This paper presents timing proficient and minimal effort equipment customization of a single center broadly useful processor architecture for Montgomery Modular Multiplication (MMM) and exponentiation. MMM requests a ton of calculation time and is the key for effective cryptography applications. Our mixture approach comprises in elaborating an enhanced C code (programming) while at the same time customizing a Tensilica® Xtensa processor instruction set architecture in request to quicken MMM.

W. Lin et al.,[4] Montgomery modular multiplication is generally utilized in open key cryptosystems. This work tells the best way to loosen up the information reliance in regular word-based calculations to expand the likelihood of reusing the present expressions of factors. With the extraordinarily loosened up information reliance, we at that point proposed a novel scheduling plan to mitigate the quantity of memory access in the created adaptable architecture. Scientific outcomes demonstrate that the memory data transmission necessity of the proposed adaptable architecture is just about 1/(w - 1) times that of ordinary versatile architectures, where w indicates word size. The proposed one additionally retains an inertness of precisely one cycle between the activities of similar words in two back to back emphasess of the Montgomery modular multiplication calculation when employing enough processing components.

J. Ye, et al., [5] Montgomery modular multiplication is broadly utilized in open key cryptosystems. This paper introduces a vitality effective architecture for word-based Montgomery modular multiplication calculation. Using the proposed architecture mapping plan in reliance chart, the switching movement of bit can be incredibly diminished. Furthermore, the proposed plan likewise retains one-cycle inactivity between neighboring processing components. Trial results dependent on TSMC 90-nm CMOS innovation demonstrate that contrasted with the related work, the proposed structure accomplishes about 17% decrease in vitality utilization for completing one 1024-piece Montgomery modular multiplication.

Table 1: Summery of literature review

Sr No.	Author Name	Publish Details	Proposed Work	Outcome
1	S. S. Erdem	IEEE 2017	Proposed Montgomery algorithm and its complexity	Logic formulas for the bits of the pre computation - θ^{-1} mod 2^{δ} used
2	M. Morales	IEEE 2016	Scalable hardware architecture for modular multiplication in prime fields	Throughput of 242 Mbps using only 219 FPGA slices
3	Q. Yang	IEEE 2015	Non inter leaved systolic secure architectures for MMM	Throughput rate of its MFM is 34.44% higher
4	R. Dou	IEEE 2014	Block-level parallel algorithm for MM	Maximizes the speedup ratio
5	S. Kuang	IEEE 2013	Proposed Modular exponentiation	60% energy saving and 24.6% throughput improvement

III. Montgomery Modular Multiplication Algorithms

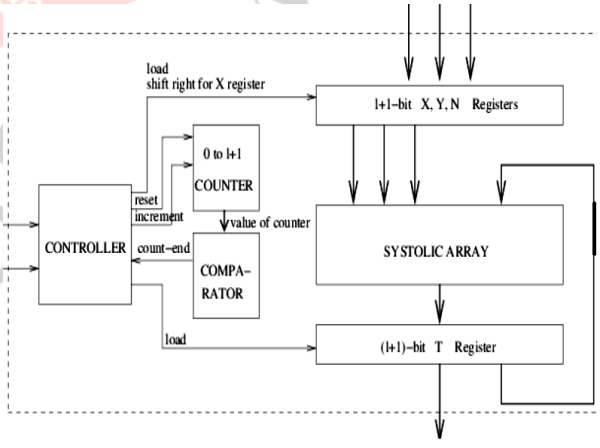

Figure 3: Architecture of the Montgomery modular multiplier circuit

It is expect m = {mn−1 mn−2… m0} is standardized, that is ½ d ≤ mn−1 < d or ½ d n−1 ≤ m < d n. It is ordinarily the

situation with RSA moduli. If not, we need to standardize it: supplant m with 2km. A modular decrease step (examined beneath) fixes the outcome: having Rk = a mod 2km determined, Rk − q•m, where q is processed from the leading digits of Rk and 2km. These de/standardization steps are just performed toward the beginning and end of the figurings (if there should be an occurrence of an exponentiation chain), so the amortized expense is irrelevant.

There are an essentially 4 calculations utilized in memory constrained, digit sequential applications (keen card, secure co-processors, shopper gadgets, and so on.): Interleaved push multiplication and decrease, Montgomery, Barrett and Quisquater multiplications.

A. Montgomery multiplication

It is straightforward and quick, utilizing ideal to-left divisions. Toward this path there are no issues with conveys (which engender far from the handled digits) or with estimating the remainder digit wrong, so no amendment steps are vital. This gives it some 6% speed favorable position over Barrett's decrease and over 20% speed advantage over division based decreases. The customary Montgomery multiplication figures the item in "push request", yet despite everything it can exploit a speedup for squaring. The main impediment is that the numbers must be changed over to a unique structure before the counts and fixed toward the end, that is, noteworthy pre- and post-processing and additional memory is required.

The item Stomach muscle can be determined interleaved with the decrease, called the Montgomery multiplication. It needs squaring-speedup as noted previously. The instruction x =(x+aib+t•m)/d is a circle through the digits of B and m from appropriate to left, keeping the convey propagating to one side.

B. RSA Algorithmic

RSA involves an open key and a private key. The open key can be known by everybody and is utilized for encrypting messages. Messages scrambled with the open key must be unscrambled in a sensible measure of time using the private key. The keys for the RSA calculation are created the following way:

1. Choose two distinct prime numbers p and q. For security purposes, the integer's p and q ought to be picked indiscriminately, and ought to be of comparative piece length.

2. Compute n = pq. n is utilized as the modulus for both people in general and private keys. Its length, normally communicated in bits, is the key length.

3. Compute φ(n) = φ(p)φ(q) = (p − 1)(q − 1) = n - (p + q - 1), where φ is Euler's totient work.

4. Choose an integer e with the end goal that 1 < e < φ(n) and gcd(e, φ(n)) = 1; i.e., e and φ(n) are co prime. e is discharged as the open key example .e having a short piece length and little Hamming weight brings about progressively effective encryption – most usually 216 + 1 = 65,537. Be that as it may, a lot littler estimations of e, (for example, 3) have been demonstrated to be less secure in certain settings.

5. Determine d as d ≡ e−1 (mod φ(n)); i.e., d is the multiplicative inverse of e (modulo φ(n)). This is all the more obviously expressed as: understand for d given d·e ≡ 1 (mod φ(n)). This is frequently figured using the all-inclusive Euclidean calculation. Using the pseudocode in the Modular integers segment, inputs an and n relate to e and φ(n), individually. d is kept as the private key example.

The open key comprises of the modulus n and general society (or encryption) type e. The private key comprises of the modulus n and the private (or decoding) type d, which must be stayed discreet. p, q, and φ(n) should likewise be stayed discreet in light of the fact that they can be utilized to figure d.

V. CONCLUSION

Montgomery Modular multiplier demonstrated to be productive for the situation of region just as timing constraints. In any case, one more task of multiplication and modular activity must be finished. In the parallel activity, for each Montgomery modular multiplier there is extra task for multiplication and modular task, this will diminish the clock cycle just as area in the chip

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]. 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.

[3]. E. C. Culau, G. C. Marchesan, N. R. Weirich and L. L. de Oliveira, "An Efficient Single Core Flexible Processor Architecture for 4096-bit Montgomery Modular Multiplication and Exponentiation," *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, Florence, 2018, pp. 1-5.

[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]. 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.

[6]. 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.

[7]. A. Rezai and P. Keshavarzi, "High-Throughput Modular Multiplication and Exponentiation Algorithms Using Multibit-Scan–Multibit-Shift Technique," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 23, no. 9, pp. 1710-1719, Sept. 2015.

[8]. 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.

[9]. 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.

[10]. M. Knezevic, F. Vercauteren, I. Verbauwhede, "Faster interleaved modular multiplication based on Barrett and Montgomery reduction methods", IEEE Trans. Comput., vol. 59, no. 12, pp. 1715-1721, Dec. 2010.

[11]. A. Miyamoto, N. Homma, T. Aoki, A. Satoh, "Systematic design of RSA processors based on high-radix Montgomery multipliers", IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 19, no. 7, pp. 1136-1146, Jul. 2011.

[12]. A. Rezai, P. Keshavarzi, "High-throughput modular multiplication and exponentiation algorithms using multibit-scan–multibit-shift technique", IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 23, no. 9, pp. 1710-1719, Sep. 2015.

[13]. O. Arazi, H. Qi, " On calculating multiplicative inverses modulo", IEEE Trans. Comput., vol. 57, no. 10, pp. 1435-1438, Oct. 2008.

[14]. T. Yanık, E. Savaş, Ç. K. Koç, "Incomplete reduction in modular arithmetic", IEE Proc.-Comput. Digit. Techn., vol. 149, no. 2, pp. 46-52, Mar. 2002.