International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 12, Issue 01, January 2023)

Montgomery Modular Multiplication (MMM) for FPGA-VLSI 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*— This paper presents review of performance of Montgomery modular multiplication algorithm using VLSI architecture. The Montgomery algorithm is a fast modular multiplication method frequently used in cryptographic applications, in which the efficiency of cryptosystem depends on the speed of modular multiplication. This study provides the comparison between different modifications done in Montgomery modular multiplication.**

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

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. This method dodges the unpredictability of remainder digit estimation and amendment found in standard division calculations.

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.

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.

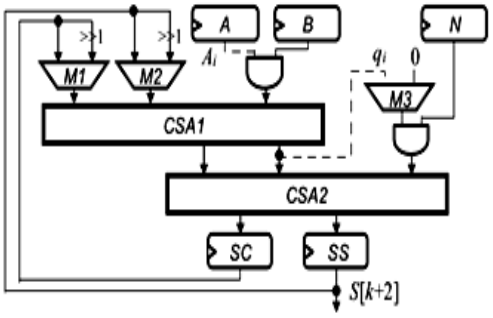

Figure 1: MMM

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. LITERATURE SURVEY

A. A. H. Abd-Elkader et al.,[1] This brief presents FPGA-based upgraded execution of Montgomery Modular Multiplier (MMM) design. The clever engineering of the proposed plan upgraded the greatest recurrence of the plan and furthermore the involved region on the designated FPGA.

A Xilinx Virtex-6 FPGA execution of the proposed engineering contrasting and other related plans uncovered that, our plan possesses the littlest region, and the proficiency is upgraded in the reach between 1.2 to 11.7 times the productivity of other significant plans. The proposed plan is executed as a modular multiplier for lightweight elliptic bend cryptography (ECC) over broad GF(p). The proposed design is designated the equipment execution of lightweight cryptographic modules utilized on the Framework on Chip (SoC) and Web of Things (IoT) gadgets.

S. M. - H. Farzam et al.,[2] present a pipelined Montgomery multiplier customized for SIKE primes. The inertness of this multiplier is far more limited than that of the past work while its recurrence contends with the most elevated appraised ones. The execution results on a Virtex-7 FPGA show that this multiplier works on the time, the region time item (AT), and the throughput of processing modular increase by something like 2.30, 1.60, and 1.36 times over SIKE primes separately. We have additionally fostered a central processor like design to perform SIDH and SIKE utilizing a few cases of our modular multiplier. Utilizing four multipliers on a Virtex-7 FPGA, the exemplification and the decapsulation of SIKE can be performed something like 1.45 times quicker while working on the No less than 1.35 times over all SIKE primes.

F. Pajuelo-Holguera et al.,[3] This article subtleties a quick and proficient execution of the Montgomery Modular Increase by exploiting equal multipliers and adders. This execution was modified in significant level union language and tried on a FPGA gadget. To test the exhibition of the proposition, a consecutive variant of the calculation was likewise executed in equipment. Besides, we contrasted the equal execution and a product form and with five commitments from the writing. Along these lines, we found that our proposition works on the presentation of any remaining executions.

M. - H. Farzam et al.,[4] Public key cryptography lies among the main bases of safety conventions. The exemplary occurrences of these cryptosystems are as of now not secure when an enormous scope quantum PC arises. These cryptosystems should be supplanted by post-quantum ones, for example, isogeny-based cryptographic plans. Supersingular isogeny Diffie-Hellman (SIDH) and key exemplification (SIKE) are two of the main such plans. To work on the exhibition of these conventions, we have planned a few modular multipliers. These multipliers have been executed for every one of the excellent fields utilized in SIKE cycle 3, on a Virtex-7 FPGA, showing a time and region time item improvement of up to 60.1% and 64.5%, separately.

These multipliers are likewise reasonable for applications like RSA, as shown by executions for 512-digit, 1024-cycle, and 2048-piece conventional moduli on a Virtex-7 FPGA. Our quickest multiplier has been utilized in the execution of SIDH and SIKE cycle 3. Utilizing six cases of this multiplier, SIDH finishes after 7.33, 8.93, 13.39, and 18.67 milliseconds and the exemplification and the decapsulation of SIKE is acted in 7.13, 8.68, 13.08, and 18.16 milliseconds over p 434 , p 503 , p 610 , p 751 , separately, which yields a least improvement component of 1.23.

B. Zhang et al.,[5] The proposed in this is a versatile high-radix (i.e., 2m) Montgomery Modular (MM) Duplication circuit supplanting the whole number increases in every cycle of the Montgomery MM calculation (connected with the result of m pieces of the multiplier and the multiplicand) with convey save compressions and totally taking out expensive increases. Besides, the proposed Montgomery MM disintegrates the actual multiplicand utilizing a radix of 2w with $w \geq 2m$, consequently accomplishing a versatile plan, which can convey an issue dormancy of one cycle and a cycle (count) idleness of O(N2/(wmp)) where p signifies the quantity of accessible handling components, every one of which is intended to finish the above emphasis by registering to a limited extent the result of w pieces of the multiplicand and m pieces of the multiplier. The region intricacy of the proposed Montgomery MM is O(wmp) , and consequently, the Region Inactivity Item intricacy is O(N2) .

J. Ding et al.,[6] In this concise, a non-least certain structure (NLP) based modular duplication technique that joins Karatsuba and textbook augmentation is applied in Montgomery modular increase, which saves 2 base increases contrasted with Karatsuba-just plans and permits pipeline construction to utilize the parallelism in enormous modular augmentations. In view of this strategy, 256-cycle and 512-bit modular multipliers are built with 3-way and 4-way NLP multipliers on FPGA stage. Carried out on Virtex-6, the 256-cycle configuration can play out a modular duplication in 62.6 ns and just requires 3.5K LUTs and 24 DSPs, which displays low-dormancy and minimal expense among past works.

G. Gallin et al.,[7] In this paper, we present modular multipliers for equipment executions of (hyper)- elliptic bend cryptography on FPGAs. The great modulus P is nonexclusive and can be designed at run-time to give adaptable circuits. A finely-pipelined engineering is proposed for covering the fractional items and decreases steps ready to go of designed DSP cuts. For example, 2, 3, or 4 free duplications can share the equipment assets simultaneously to cover inward latencies.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 12, Issue 01, January 2023)

We planned a device, dispersed as open source, for producing VHDL codes with different boundaries: width of operands, number of consistent multipliers per actual one, speed or region enhancement, conceivable utilization of BRAMs, target FPGA. Our modular multipliers lead to, at any rate, 2 times quicker as well as 2 times less circuits than cutting edge administrators.

Z. Gu et al.,[8] Toom-Cook duplication is a hypothetically more productive duplication calculation than customarily utilized Karatsuba and textbook increase yet is seldom utilized in functional equipment plans because of its intrinsic careful divisions, which are time-consuming and challenging for equal and sequential speed increase. This brief proposes a technique for sans division Toom-Cook duplication based Montgomery modular increase, which makes it feasible for Toom-Cook augmentation to be applied in viable and productive equipment executions. We likewise give an equipment execution of modular multipliers of 256 pieces and 1024 pieces with benefits on region time-item over past explores.

S. S. Erdem et al.,[9] The Montgomery calculation is a quick modular duplication strategy oftentimes utilized in cryptographic applications. This paper explores the digit-sequential executions of the Montgomery calculation for enormous whole numbers. An itemized investigation is given and a tight upper headed is introduced for the middle outcomes got during the digit-sequential calculation. In view of this examination, a productive digit-sequential Montgomery modular multiplier design utilizing convey save adders is proposed and its intricacy is introduced. In this engineering, pipelined convey select adders are utilized to perform two last undertakings: adding convey save vectors addressing the modular item and taking away the modulus from this option, on the off chance that further decrease is required.

W. Dai, et al.,[10] The modular augmentation activity is the most time-consuming activity for number-hypothetical cryptographic calculations including enormous numbers, like RSA and Diffie-Hellman. Executions uncover that in excess of 75% of the time is spent in the modular duplication work inside the RSA for more than 1,024-piece moduli. There are quick multiplier designs to limit the postponement and increment the throughput utilizing parallelism and pipelining. Anyway such plans are enormous with regards to region and low in effectiveness. In this paper, we coordinate the quick Fourier change (FFT) strategy into the McLaughlin's system, and present a better FFT-based Montgomery modular augmentation (MMM) calculation accomplishing high region time proficiency.

D. D. Chen et al.,[11] presents modular augmentation is the center activity openly key cryptographic calculations like RSA and the Diffie-Hellman calculation. The productivity of the modular multiplier assumes a urgent part in the exhibition of these cryptographic techniques. In this paper, enhancements to FFT-based Montgomery Modular Duplication (FFTM3) utilizing convey save number juggling and pre-calculation strategies are introduced. Besides, pseudo-Fermat number change is utilized to enhance the upheld operand sizes for the FFTM 3 . The asymptotic intricacy of our strategy is O(l log l log l), which is equivalent to the Schonhage-Strassen increase calculation (SSA).

III. Modular Arithmetic

As a quick review, ***r*mod*n*** is equal to the remainder when we divide *r* by *n*. Addition, subtraction, and multiplication in modular arithmetic obey two basic rules.

1. If $a + b = c$, then $(a + b) \bmod n$ is congruent to $c \bmod n$.
2. If $a \bmod n$ is congruent to $d \bmod n$ and $b \bmod n$ is congruent to $e \bmod n$, then $(a + b) \bmod n$ is congruent to $d \bmod n + e \bmod n$.

In each of these rules, the plus sign can be replaced by a subtraction or multiplication sign. These rules state that we can first perform the operation and then find that number mod*n*, or we can find each of the numbers mod*n* and then perform the operation on them. It's important to note that when dealing with subtraction, you may get negative numbers. When this happens, you add multiples of the modulus *n* until you get a number between 0 and *n* - 1.

Modular multiplication is pretty straightforward. It works just like modular addition. You just multiply the two numbers and then calculate the standard name. For example, say the modulus is 7.

Example 1:

Find the remainder of 15 x 17 x 19 when divided by 7.

Solution:
On dividing 15 by 7 we get 1 as remainder.
On dividing 17 by 7 we get 3 as remainder.
On dividing 19 by 7 we get 5 as remainder.

Remainder of the expression (15 x 17 x 19)/7 will be equal to (1 x 3 x 5)/7.

Combined remainder will be equal to remainder of 15/7 i.e. 1.

Application-

- Modular arithmetic has many applications in cryptography and computer science.

- It's often used to detect errors in identification numbers.
- Think about the kinds of identification numbers we use every day. Credit cards, bank accounts, and product barcodes all involve long strings of numbers.
- Modular arithmetic is used extensively in pure mathematics, where it is a cornerstone of number theory.

IV. Conclusion

Modular multiplication of long integers is an important building block for cryptographic algorithms or the other hand Montgomery Algorithm for modular multiplication with a large modulus has been widely used in public key cryptosystems for secured data communication. MMM is turned out to be proficient on account of region just as timing limitations. In any case, one more task of multiplication and modular activity must be finished. In future we design of MMM using the Xilinx software for FPGA-DSP applications.

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.