High-Speed Triple Operand Adder for Graphic processing and Cryptography

PREMANANTHAN G M.E.,(Ph.D.).
Assistant Professor / ECE
Karpagam College of Engineering
Coimbatore, India
premananthan.g@gmail.com

GOPI P
Electronics and Communication Engineering
Karpagam College of Engineering
Coimbatore, India
20l214@kce.ac.in

JAGATHESH V
Electronics and Communication Engineering
Karpagam College of Engineering
Coimbatore, India
20l218@kce.ac.in

KAVISWARAN S
Electronics and Communication Engineering
Karpagam College of Engineering
Coimbatore, India
20l222@kce.ac.in

ALAGU THANUSH A
Electronics and Communication Engineering
Karpagam College of Engineering
Coimbatore, India
20l203@kce.ac.in

Abstract- ***consumer electronics sector has led to a sharp increase in demand for high-performance, low-power adders with big operands. Many operands require a lot of energy when using standard fast adder designs like parallel prefix adders. Hybrid designs offer a viable solution to the problem of striking a balance between power consumption and latency in large operand addition. The least significant carriers in large parallel-prefix adders are generated significantly faster than the most significant ones; this insight is capitalized upon in this novel hybrid adder architecture for enormous operands. Due to the lack of significant impact on overall performance, the authors have chosen not to include fast ideas relating to this issue when applying carries to the final computing of the least-significant bits. Thus, without sacrificing speed, this method minimizes the size of the summing blocks at the least important locations. In order to further minimize latency, the suggested adder's carrier network generates and distributes the complement of the carries. Comparing this proposed adder to the most advanced adders for big operands, VLSI implementation findings utilizing 45-nm-TSMC technology show that it offers a reduced area-delay product and consumes less energy.***

Keywords: Reduce area-space, high-speed, parallel-prefix adders, low-power.

I INTRODUCTION

In order to maintain physical security and maximise system performance, a hardware implementation of cryptographic algorithms must be established. Modular arithmetic is used extensively in several cryptographic techniques to perform a number of mathematical operations, including modular addition, modular multiplication, and modular exponentiation. As such, the use of congruential modular arithmetic operations is critical to the effective implementation of cryptographic algorithms. The Montgomery method, which uses a three-operand binary addition, is the recommended option for effectively completing modular multiplication and exponentiation. In pseudo-random bit generators (PRBGs) such as the linked variable input LCG (CVLCG), modified dual-LCG (MDCLCG), and connected LCG (CLCG), this basic arithmetic operation is used. MDCLCG stands out as the most secure alternative among these.

II LITERATURE SURVEY

[1] In this work, we provide an ECC processor for 256-bit point multiplication on the Edwards25519 twisted Edwards curve that is both fast and small in area, and that is immune to side-channel assaults. The CPU uses special hardware designs to complete tasks like point addition and doubling, which need just 516 and 1029 clock cycles, respectively. The method is fast and does not compromise security while enabling rapid scalar multiplication with minimal hardware consumption. The CPU is crucial for network security and cryptography.

[2] Digital signatures using elliptic curve cryptography (ECC) are often used to secure communication. This work uses a hardware-software strategy to demonstrate an effective dual-field ECC processor. The processor employs a Modular Arithmetic Logic Unit (MALU) for effective modular arithmetic operations and supports any elliptic curve. The processor offers a high level of hardware efficiency and flexibility and can carry out a variety of operations using multiple algorithms. One-point multiplication takes between 0.60ms, and 6.75ms to implement in FPGA and ASIC.

[3] This study provides an easy-to-use and practical Montgomery multiplication method for carrying out the Montgomery modular multiplication with good performance and economy. The suggested multiplier employs a sole-tier carry-save adder (CSA) to prevent the propagation of carries during addition operations and works with data that is given in binary format. By recalculating operands and converting the carry save format to binary, the CSA (Carry Save Adder) performs a dual function that lowers hardware costs and shortens the critical path delay. But in order to perform modular multiplication, it adds extra clock cycles. A configurable CSA (CCSA) is suggested as a solution, cutting down on format conversion and operand precomputation clock cycles by half. To further minimize critical path time, the system recognizes and avoids superfluous carry-save addition operations. With this strategy, throughput is increased while the effects of additional clock cycles are minimized. In comparison to earlier designs, experimental

results demonstrate better performance and area-time efficiency.

[4] To reduce point doublings by half, this study looks at the computation of ECDSA signature verification on a twisted Edwards curve. The 207-bit prime field Fp is covered by a curve that is the focus. A tiny CPU in a 0.13 _m CMOS ASIC for IoT applications and a quick signature verification architecture employing FPGA acceleration for server-side applications are presented as two different hardware designs. The designs include different trade-offs and optimizations, which makes them useful for IoT applications.

[5] To decrease the consumption of energy and Montgomery modular multipliers throughput increases, this research suggests an energy-efficient method and design. The suggested design avoids carry write and save addition operations in registers, which reduces energy use and increases throughput. Additionally, the authors lower the energy consumption of the storage element by modifying the barrel register full adder (BRFA) to take advantage of gated clock design approaches. According to experimental findings, these methods can increase throughput for a 1024-bit Montgomery multiplier by 24.6% while saving up to 60% energy.

[6] To improve data security in cryptography applications during data transmission and storage, the paper provides an improved dual-CLCG approach. By producing pseudo-random bits at a steady pace to reach the longest sequence possible, this technique reduced the issues related to initial clock delay and memory use. With only one starting clock delay, the enhanced PRBG approach minimizes hardware complexity and clears all 15 benchmark tests NIST as well as the extreme period test of 2n. The architecture on a commercial FPGA chip can be implemented by Verilog-HDL.

[7] By offering a range of well-crafted formulations and introducing a novel family of parallel prefix adders at the architectural level, this paper advances our understanding of parallel prefix adders. These adders show an impressive up to 26% improvement in area-throughput performance over state-of-the-art alternatives, while still maintaining speed parity with existing models.

[8] Because of its difficult prime factorization qualities, the Blum Blum Shub(BBS) Pseudorandom Bit Generator (PRBG) provides strong cryptographic security for Internet of Things applications. However, the lengthy modular multiplication of large numbers is mostly responsible for its processing cost. The utilization of the Montgomery approach introduces both critical path time and clock latency, albeit being a viable option. Our investigation proposes a revised radix-2 iterative Montgomery modular multiplier as a resolution to address this issue. In the case of a 1024-bit BBS generator, this solution involves substituting double two-operand adders with a singular three-operand adder. We used Verilog HDL to implement this design on a Virtex5 FPGA chip, and the end result was a significant 93.87% reduction in overall latency.

In [9] Grounded on a revolutionary serial digit processing approach, the Montgomery modular multiplication architecture presented in our work is both original and efficient. Thanks to this design, which converts high-radix multiplication partially into multiplication in binary, several zero bit multiplications may be completed in a single clock cycle. Our design stands out for its superior performance over most existing solutions when run on a Xilinx Virtex 5 FPGA, providing faster computation times and greater throughput rates.

III PROPOSED WORK

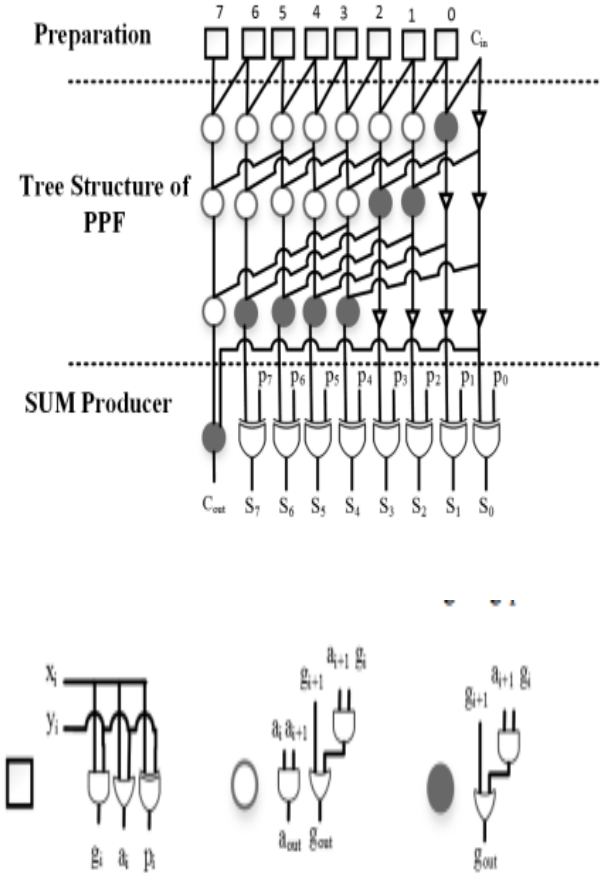


Fig.1 Three-operand adder structure with an 8-bit wide band

The four main steps of the recommended adder's setup are base, PG logic, bit addition, and logic of sum. The quantity of stages prefix in the PG logic has a major impact on the adder's efficiency. Thus, the following elements must be looked at in the suggested architecture of a three-operand adder calculating the maximum propagation gate delay

$$\begin{aligned} T_{prop} &= T_{bitadd} + T_{base} + T_{PG} + T_{sum} \\ &\approx 2T_X + T_X + 2\lceil \log_2 n' + 1 \rceil T_G + T_X \\ &\approx 4T_X + 2\lceil \log_2 n' + 1 \rceil T_G \end{aligned}$$

(TProp):

Similarly, the area of the proposed adder hardware (Aprop) can be estimated as:

$$A_{prop} = A_{bitadd} + A_{base} + A_{PG} + A_{sum}$$
$$\approx (2nA_X + 3nA_G) + (n+1)(A_X + A_G) + \left[2n + 3s\left\lceil\frac{n}{2}\right\rceil - 3\times 2^s + 3\right]A_G + nA_X$$
$$A_{prop} \approx (4n+1)A_X + \left[6n + 3s\left\lceil\frac{n}{2}\right\rceil - 3\times 2^s + 4\right]A_G$$

Here, s =log2 n –1and

For a thorough evaluation, the extension of the 'hybrid Han-Carlson two-operand adder concept to formulate a three-operand adder architecture is introduced', where n=n−1. The assessment of the first-order timing-area complexity for this extended architecture, termed the Hybrid Han-Carlson three-operand adder (HHC3A), is conducted as follows:

$$T_{HHC3A} = T_{bitadd} + T_{base} + T_{PG} + T_{sum}$$
$$\approx 2T_X + T_X + 2\lceil\log_2 n^* + 2\rceil T_G + T_X$$
$$\approx 4T_X + 2\lceil\log_2 n^* + 2\rceil T_G$$
$$A_{HHC3A} = A_{bitadd} + A_{base} + A_{PG} + A_{sum}$$
$$\approx (2nA_X + 3nA_G) + (n+1)(A_X + A_G) + (5n - 3\log_2 n - 3)A_G + nA_X$$
$$A_{HHC3A} \approx (4n+1)A_X + (9n - 3\log_2 n - 2)A_G$$

Here, 5 is subtracted from n to get the value of n_ for n _ 8. It also expands "the ultrafast two-operand adder, as described in [20]" to an adder which is three-operand (UF3A), and evaluates the complexity area-time.

A comparison with current designs, including CS3A, HC3A, HHC3A, and UF3A, is provided together with the area and time complexity analysis of the recently announced three-operand adder architecture. In comparison to the HC3A adder design, the findings show a considerable reduction in AG gate area of 43.4%, 50.2%, and 55.6% for operand widths of 32, 64, and 128 bits, respectively, coupled with a concurrent reduction in critical latency. Moreover, for sizes of 32, 64, and 128 bits, the critical latency of the proposed adder is significantly lower than that of the CS3A adder, namely by 81.2%, 89.1%, and 93.7%.

Interestingly, while being two gates slower, the hybrid Han-Carlson-based HHC3A adder has a gate area consumption that is 10% to 18% lower than that of the suggested adder. However, at the expense of a gate area nearly twice as large as that of the HHC3A and the recommended three-operand adders, the ultra-fast two-operand-based UF3A is the fastest three-operand adder, outperforming the proposed adder by three gates.

IV REASON FOR PROPOSING

To enhance both area efficiency and speed in the proposed design, several strategies are implemented:

(i) To increase space efficiency, the proposed architecture incorporates two distinct types of units for sum calculation. It selects the Sum Producer type 2, which possesses a unique configuration resembling a carry choose adder. We utilize CSL adders due to their speed, as their latency is comparable to that of a single multiplexer (MUX) and is significantly lower than that of other varieties. While CSL adders are widely recognized for their rapidity, their increased area requirement is often perceived as a compromise.

The proposed design strategically incorporates several types of sum producer blocks, setting it apart from solutions that use a uniform structure for all of them. When the input carry bit's complement, $\bar{c}i$, becomes available sooner, the Sum Producer type 1 is strategically employed. There is a reduction in area overall since this kind has a more definite structure than the second type. The suggested adder achieves quicker final addition results for type 1 blocks even though they have more latency because the carry-in bits' complement is available sooner in the calculation.

(ii) The suggested adder generates and propagates carry complement $\bar{c}$ i rather than using ci in order to reduce latency. In this method, $\bar{c}$ i · $\bar{c}$ i is evaluated instead of ci, and the generate complement ($\bar{g}$ i) and kill signal (ki = $\bar{a}$ i· $\bar{b}$ i) are utilized instead of the generate (gi) and alive (ai) signals that are often employed in regular PPF systems. In the Preparation level, the circuit speed is higher than in normal PFF designs because NOR and NAND gates are used in the calculation of ki and $\bar{g}$ i for each bit position. Furthermore, gates NAND/NOR are included in the anticipated adder to reduce, particularly in the path which was critical.

Suppose A = a – 1an − 2…a0 and B = bn – 1bn − 2…b0 be two examples of input n-bit operands. The complemented values of the carriers ($\bar{c}$ i′s) are calculated using the following formulas as a basis:

$$ki = \bar{a}i \cdot \bar{b}i = \overline{ai + bi}$$
$$\bar{g}i = \overline{ai \cdot bi}$$

Since ci + 1 = ai ·bi + ci . ai + bi , $\bar{c}$ i + 1 can be calculated as follows:

$$\overline{ci} + 1 = ai \cdot bi + ci \cdot ai + bi = ai \cdot bi \cdot ci \cdot ai + bi$$
$$= ai \cdot bi \cdot ci + ai \cdot bi$$
$$= ai \cdot bi \cdot ci + ai \cdot bi \cdot ai + bi$$
$$= ai \cdot bi \cdot ci + ai + bi$$
$$= ai \cdot bi \cdot ci + ai + bi \cdot ai \cdot bi$$

Therefore:

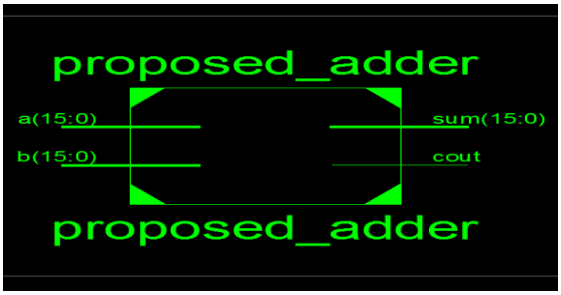
$$\bar{c}i + 1 = \bar{g}i\bar{c}i + ki$$

The nodes configuration employed in the planned proposal is depicted in Figure. 1. It is important to note that the black node and the starred node serve identical functions. However, due to a higher power consumption associated with the starred node, it is selectively utilized only in the critical path to enhance latency.

V PERFORMACE ANALYSIS:

S.No	Parameter	Existing Method	Proposed Method
1	Luts	16	55
2.	Flip Flops	16	55
3	Power in mW	413.23	206.49
4.	Combinational Delay in ns	1.322	2.993

V RESULT



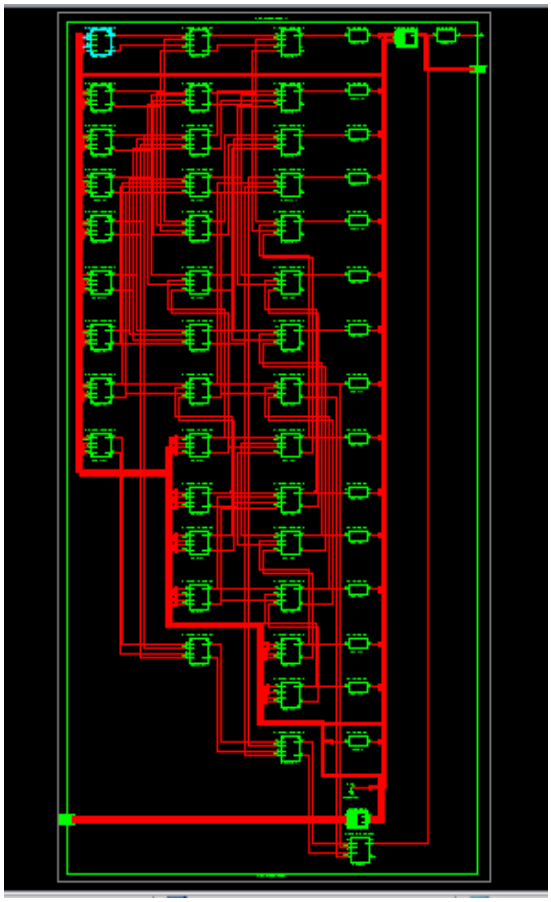


Fig.2 RTL diagram of proposed TOA

A) DEVICE UTILIZATION SUMMARY

```
Device utilization summary:
---------------------------

Selected Device : 7v585tffg1157-3


Slice Logic Utilization:
 Number of Slice LUTs:                   55  out of  364200     0%
    Number used as Logic:                55  out of  364200     0%

Slice Logic Distribution:
 Number of LUT Flip Flop pairs used:     55
   Number with an unused Flip Flop:      55  out of     55   100%
   Number with an unused LUT:             0  out of     55     0%
   Number of fully used LUT-FF pairs:     0  out of     55     0%
   Number of unique control sets:         0

IO Utilization:
 Number of IOs:                         49
 Number of bonded IOBs:                 49  out of    600     8%
```

B) POWER SUMMARY

```
2.3.  Power Supply Summary
-------------------------------------------------------------
|                Power Supply Summary                       |
-------------------------------------------------------------
|                       | Total  | Dynamic | Quiescent |
-------------------------------------------------------------
| Supply Power (mW)     | 206.49 | 0.00    | 206.49    |
-------------------------------------------------------------
```

C) TIMING SUMMARY

```
Timing Summary:
---------------
Speed Grade: -3

   Minimum period: No path found
   Minimum input arrival time before clock: No path found
   Maximum output required time after clock: No path found
   Maximum combinational path delay: 2.993ns
```

D) SIMULATION WAVE FORM

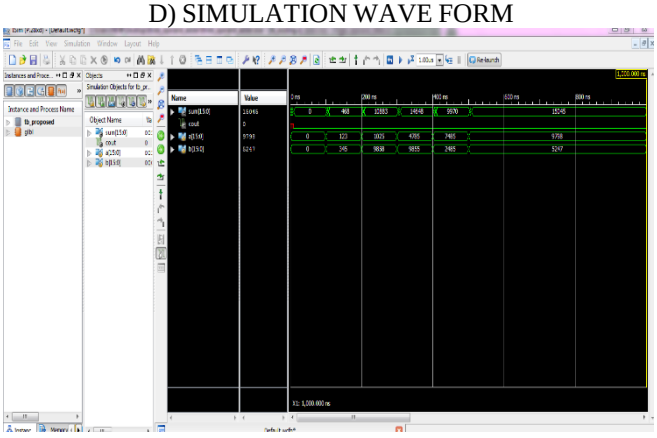


Fig.3 Output Waveform

VI CONCLUSION

This creative design, which introduces a new adder structure, uses modified sum producer blocks and PPF as its fundamental building blocks and is well suited to effectively handle big operands. To improve overall performance while also saving hardware costs, this article describes two different iterations of these essential components.

The first type of sum-producer block is more affordable and has a simpler architectural form; this is mainly because the carry-in's complement was available earlier. The optimized CSL design is used by the second version of the sum producer, on the other hand. This clever architectural idea not only increases efficiency but also makes carry-bit complement creation and transmission easier.

After extensive testing, the results unambiguously show that the proposed 16-bit adder can raise ADP (Average Delay Power) by about 5% and raise energy consumption by about 12%, indicating that it can be a viable solution for managing large operands at a reasonable cost.

VII REFERENCES

[1] "FPGA implementation of a high-speed area-efficient processor for elliptic curve point multiplication over a prime field," IEEE Access, vol. 7, pp. 178811-178826, 2019. M. M. Islam, M. S. Hossain, M. K. Hasan, M. Shahjalal, and Y. M. Jang.

[2] "Elliptic curve cryptography with efficiently computable endomorphisms and its hardware implementations for the Internet of Things," IEEE Trans. Comput., vol. 66, no. 5, pp. 773–785, May 2017; Z. Liu, J. GroBschadl, Z. Hu, K. Jarvinen, H. Wang, and I. Verbauwhede.

[3] Low-Cost High-Performance by Shiann-Rong Kuang, Kun-Yi Wu, and Ren-Yao Lu Montgomery Modular Multiplication Using VLSI Architecture, 2015.

[4] Zhe Liu, Johann Großschadl, Zhi Hu, Kimmo Jarvinen, Husen Wang, and Ingrid Verbauwhede are among the authors of the study. The Internet of Things: Hardware Implementations of Curve Cryptography with Efficiently Computable Endomorphisms, 2016

[5] Shiann-Rong Kuang, Jiun-Ping Wang, Kai-Cheng Chang, and Huan-Wei Hsu, Energy-Efficient High-Throughput MontgomeryModular Multipliers for RSA Cryptosystems, 2012.

[6] Amit Kumar Panda and Kailash Chandra Ray, "A Coupled Variable Input LCG Method and its VLSI Architecture for Pseudorandom Bit Generation", IEEE Transactions on Instrumentation and Measurement, volume 69, issue 4, 2020.

[7] AdderKumarSambhav Pandey, Dinesh Kumar B, Neeraj Goel An Ultra-Fast Parallel Prefix

[8] The 1024-bit Blum-Blum-Shub PRBG Architecture Design and FPGA Prototype was proposed by Amit Kumar Panda and Kailash Chandra Ray.

[9] The High-Throughput Modular Multiplication and Exponentiation Algorithms Using-Shift Technique was developed by Abdalhossein Rezai and Parviz Keshavarzi.

[10] Z. Liu, D. Liu, and X. Zou, “An efficient and flexible hardware implementation of the dual-field elliptic curve cryptographic processor,” IEEE Trans. Ind. Electron., vol. 64, no. 3, pp. 2353–2362, Mar. 2017.

[11] K. Panda and K. C. Ray, “Modified dual-CLCG method and its VLSI architecture for pseudorandom bit generation,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 66, no. 3, pp. 989–1002, Mar. 2019.

[12] Jinping Fan, Yujie Gu, Masahiro Hachimori and Ying Miao, "Signature Codes for Weighted Binary Adder Channel and Multimedia Fingerprinting" IEEE Transactions on Information Theory Year: 2021

[13] Titouan Coladon, Philippe Elbaz-Vincent, and Cyril Hugounenq, "MPHELL: A fast and robust library with unified and versatile arithmetics for elliptic curves cryptography", 2021 IEEE 28th Symposium on Computer Arithmetic (ARITH), Year: 2021.