THE THREE OPEREND BINARY ADDER HAS A HIGH-SPEED AND AREA-EFFICIENT DESIGN

[1]Derangula Vijaya Kumari, [2]K.Monica, [3]D.Chaitanya

[1]M.Tech Student,[23]Assistant Professor

Department Of ECE

MJR College of Engineering & Technology,Chittoor

Abstract:

Addition in mathematics is a fundamental ability. When it comes to designing specific DSP applications and microprocessors, VLSI technology is frequently utilised. The ability to add two binary integers is another benefit, as is the role it plays in a wide range of mathematical operations. If you utilise an additive to produce three operator-carrying adders, this project will allow you to generate three operand adderSums. Thereby, a more energy-efficient and faster localised construction is given, which makes use of number-based prediction to produce a highly compact triple binary adder (log2 n).

Keywords: Modular arithmetic and Han-carlson adders with carry saves are included in this package (HCA).

I. INTRODUCTION

It's polynomial-time unpredictable and safe because n = 32 bits. As a result, MDCLCG becomes more secure as operand size increases. Adder comparison multipliers and critical path delays are increased linearly in the modulo-2n hardware design. By using the three-operand adder correctly, the MDCLCG's performance can be enhanced.

A three-operand adder or two two-operand adders can be used to execute a binary addition. Cryptographic algorithms and PRBG approaches can both benefit from CS3A, a space-efficient 3-operand binary addition algorithm. When ripple-carrying, the CS3A's carry propagation latency is deleterious. MDCLCG and other cryptographic algorithms are utilised on IoT-based electronics. MDCLCG. While performing three-operand binary addition, Han-Carlson (HCA) prefixed two-operand adders can be utilised in simultaneously. By increasing the surface area by O(log2 n), the critical path time is halved [15]. 16 to 18-year-olds A VLSI design for three-operand binary addition must be efficient in order to fulfill this operation because of restricted hardware resources. An HCA-based 3-operand adder's gate area and propagation time were reduced using bitwise addition and carry prefix calculation logic in this study (HC3A)[2].

II. LITERATURE SURVEY

"Error Tolerant Adder Using Gate Diffusion Input," It was a collaboration between Meenu Pareek and Manish Singhal in 2016. [6] The outputs of digital VLSI circuits must be exact. Due to the fact that circuits are less error-prone, researchers now focus on building circuits that generate superior computation output. The primary objectives of this error-tolerant adder are to achieve high speed and power consumption (ETA). Based on the development logic approach, this study uses a 32-bit ETA as the foundation. Time and space requirements are greatly reduced.

The 2015 publication "Logical Effort Analysis of Various VLSI Design Algorithms" by Rommel M Anacan and Josephine L. Bagay provides an example. [7] Delay estimation is one of the most important aspects of any VLSI plan computation. It is often referred to as a component of a synchronised circuit strategy for better execution. Both of these words are not widely used to describe the display of different VLSI geographies. At the price of force utilisation, commotion edges, or plan effort adders are widely utilised in elite execution integrated circuits (IECs).

"High performance VLSI adders" was written by R. Suganya and D. Meganathan in 2015. Advanced CMOS technology will be used to design and study VLSI adders for various piece levels up to 64 cycles. Weinberger, Ling and Manchester transmit chain snake adders are preferred because of the importance of their presentation. Due to the fact that these three calculations rely heavily on power, energy, and delay, other VLSI adders were used. It consumes less power than other VLSI adders because of the reduced number of transistors.

Vinod Kumar Naik and Mohamed Aneesh Y., "The design of a low-power and high-speed VLSI carry select adder," [9] A low-power VLSI carry-select adder is implemented in this study. Mobile devices and biomedical applications typically use wireless receivers that support various protocols. The adder is the unit's most critical component.

N. Ravindran and R. Mary Lourde, "An optimal VLSI design of a 16-bit ALU," 2015. [10] Using this 16-bit ALU, we were able to improve the efficiency of VLSI design. To create the 16-bit Arithmetic Logic Unit, many logic families were combined. CMOS (Complex Metal Oxide Semiconductor) is used for fundamental logic operations, while pseudoNMOS (N-Type Metal Oxide Semiconductor) and Pass Transistor logic are used for multiplexers. This will lead to an increase in productivity. The chip level design is implemented using the schematic editor, which is also used to evaluate the design.

With the help of Narule et al., a floating point addition was built with a delay in the operand (2016). The internal width, as defined by IEEE Std 754, is significantly reliant on delays. An LZA and a compound adder are utilised to improve the overall type composition. 7.79 percent less latency than the three operand adder is possible with the proposed alternative.

To install three operands, Dave et al. recommend parallel prefix adders, an adder structure. Because it doesn't require a separate adder unit to perform three-input addition, this technique has a number of advantages.

When entering huge numbers, Tsiaras et al. (2017) suggest utilising the Logarithmic Number System (LNS). Using the data with the highest input value is the recommended method. The complexity and efficiency of this multi-operand adder library are evaluated using four alternative bit sizes, including 4, 8, 16, and 11 bits.

To conduct three-operator binary adds in a modern high-speed and efficient terminal architecture, Liu et al. (2011) designed carryprefix logic that required less space, power, and latency of less than a millisecond (log2 n).

Two low-cost, feature-identical 3D stackers developed by Voicu and colleagues in 2017 made it possible for hybrid adders to be stacked three-dimensionally. N-bit extension applied to the equivalent K-linked K executes two N / K-bit adds in each phase according to the predicted computing idea.

To swiftly add decimals to a number of binary-coded operands, Kenney and colleagues examine three different ways. When adding input operators, two algorithms predict BCD correction values and produce accurate intermediate outputs.

An entirely new approach for making repeating circuits has been developed, which Cilardo et al. (2014) describe as speculative and slow but hardly used. For an 8-bit Manchester carry chain (MCC) adder, Efstathiou et al. proposed employing a multi-output domino CMOS logic (2013). Two 4-bit chains support the charm's chains in the same way.

A new theory of latency fluctuation based on Han-Carlson parallel-prefix topology and Kogge-Stonetopology has been proposed by Esposito et al. (2015). As compared to earlier approaches, this research focuses on more accurately predicting startup add-on latency and developing a novel mistake detection network.

CMOS and transmission gate logic were employed in a 1-bit full adder presented by Bhattacharyya in 2015. An initial one-bit system eventually grew to 32. Cadence Virtuoso tools were used to implement the circuit in 180- and 90-nm technologies. A 12-bit transitionor based on low-power multiplexers was created by Jiang et al (MBA12T). Lower exchange rates, new renewable energy, and the absence of direct contact with power supply infrastructure all help to minimise short-term energy consumption. Compared to a traditional 10-transistor adder, this revolutionary add-on uses only a quarter of the power and is 64 percent faster than the usual 28-transistor CMOS add-on.

III. SYSTEM DESIGN

VLSI ARCHITECTURE OF BINARY ADDER

As seen in the diagram, a binary adder's VLSI architecture is depicted. Inputs must be organised in a logical sequence (a and b). An entirely new pre-processing stage will be introduced in the future. During the pre-processing steps, signals for propagator and generator are both generated. To propagate and create, the signals must be generated by the propagator and generator components. The carry generating unit is responsible for the generation of the grey and black cells. To do addition, the adder tree block is utilised. The final output is preserved at the post-processing stage.[4]

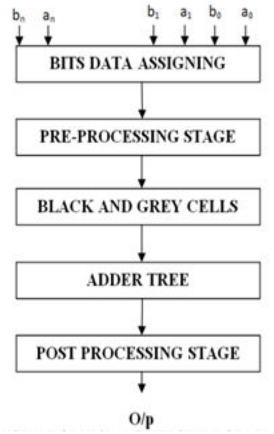


Fig.1: STRUCTURE OF VLSI ARCHITECTURE OF BINARY ADDER

Stage 1: The pre-processing stage generates signals that can be utilised to transmit and distribute information. XOR and OR gates can be implemented using the carry and generation logic of a binary adder. There is just one delay since there is only one gate for each signal.

Stage 2: The generation step of the carry is calculated using carry and bits. The entire procedure is completed in one go. The employment of intermediate signals helps to generate and propagate signals. The following equations are used to transmit and create signals. Equations are implemented in the grey and black cells[5].

Pi:j = Pi:k AND Pk-1:j
Gi:j = Gi:k OR (Pi:k AND Gk-1:j)

Stage 3: At this point, the input bits are used to execute computations. The post-processing stage is where sum and carry are calculated.

THREE OPERAND BINARY ADDER

To perform pseudorandom bit generation and other cryptography, the three-operand binary adder serves as the primary functional unit. Cryptographic algorithms must be organised on hardware in order to maximise system speed while ensuring physical security. When it comes to cryptography algorithms, modular arithmetic operations play a key role in their performance. Using three-operand binary addition as its fundamental operation, the Montgomery approach is the foundation of modular arithmetic. LCG-based pseudo random bit generators such as Coupled LCG, Modified Dual CLCG, and Coupled Variable Input LCG use three-operand binary addition as their fundamental arithmetic. Modified Dual CLCG is the most secure and random pseudo random bit generator method among the LCG-based PRBG approaches. The area and delay of this filter are inversely related to the size of the operand. VLSI architecture for a three-operand binary adder can lower the delay and size of a modified Dual CLCG[6].

A parallel prefix adder, Han-Carlson adder, is a good alternative to the carry save adder in three-operand binary adders because the ripple carry step in the latter causes substantial latency. As a result, the delay is effectively reduced while the hardware area is expanded. Parallel prefix two-operand adders have become increasingly common in recent years. Three gates quicker than the Han-Carlson, the ultra-fast adder is considered the fastest. However, the Han-Carlson adder requires twice as much gate area. A space-saving VLSI architecture and a new, high-speed, three-operand adder technology are discussed in the next section to help minimise this trade-off. Save space and time by employing an efficient VLSI design, such as a bitwise addition followed by carry prefix computation logic. Bitwise addition and carry prefix logic are the most efficient ways to accomplish a three-operand binary addition. For the parallel prefix adder, see here. Bit-addition logic, base logic, PG (propagate-and-generate) logic, and sum logic make up the prefix adder's four-step structure rather than a three-stage structure for adding three binary input operands.

Three n-bit binary inputs can be added in four steps using the new adder method. An array of full adders is used to bitwise add three n-bit binary input operands in the first stage (bit addition logic), and each full adder computes the "sum (Si)" and "carry (ci)" signals as described in. Sums can be computed using logical formulas (Si) This is accomplished by first combining the "produce" (Gi) and "propagate" (Pi) signals computed by each complete adder's "sum (Si)" and "carry" bits (base logic). When it comes to basic logic, n "squared saltire-cells" indicate the computation of Gi and Pi signals.

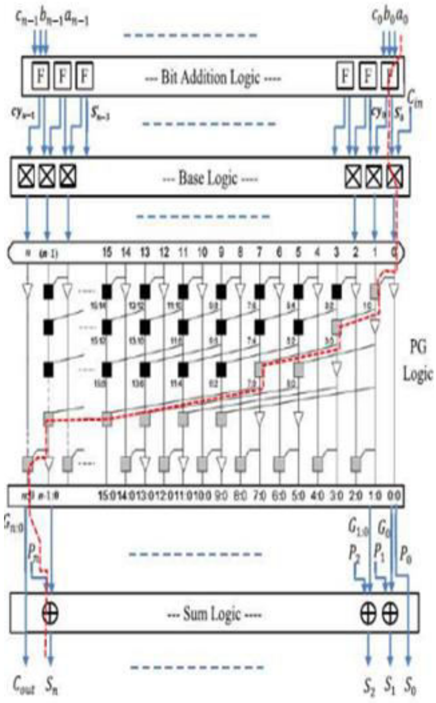


Fig 2(a) . Proposed three-operand adder; (a) First order VLSI architecture cell.

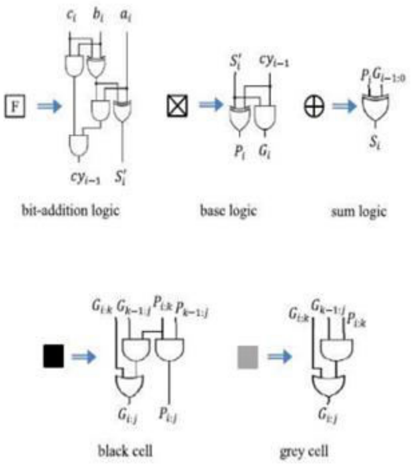


Fig 2 (b). Logical diagram of bit addition, base logic, sum logic, black-cell and grey-cell.

The external carry-input signal (Cin) is also taken into account for three-operand addition in the suggested adder technique. G0 is computed at the first saltire-cell of the base logic using S0 r Cin as an input, as shown in the diagram. As the name implies, it's an input signal with a carry signal attached to it. While generating and propagating logic (PG), a carry bit is combined using black-and-gray cell logics in the third stage. Calculating the carry signals, producing Gi j, and propagating Pi j are all shown in Figure.

MODIFIED DUAL-CLCG METHOD:

The modified dual-CLCG approach generates pseudorandom bits by congruentially modulo-2 adding the outputs of two connected linear congruential generators (CLCGs).

xi+1 ≡ a1 × xi + b1 mod2n
yi+1 ≡ a2 × yi + b2 mod2n
pi+1 ≡ a3 × pi + b3 mod2n
qi+1 ≡ a4 × qi + b4 mod2n

The pseudorandom bit sequence Zi is generated using the congruential modulo-2 equation.

$$Z_i \equiv (B_i + C_i)\mathrm{mod}2 = B_i \oplus C_i$$

Where

$$B_i = \begin{cases} 1, & \text{if } x_{i+1} > y_{i+1} \\ 0, & \text{else} \end{cases} \quad \text{and } C_i = \begin{cases} 1, & \text{if } p_{i+1} > q_{i+1} \\ 0, & \text{else} \end{cases}$$

The constant parameters are a1, b1, a2, b2, a3, b3, a4, and b4, while the starting seed values are x0, y0, p0, and q0. These numbers are used to start the simulation. The present dual CLCG technique's parameters for a maximum length period remain unchanged (as discussed in Section-II). Two independent connected outputs of the LCG are added congruently modulo-2 in the proposed improved dual-CLCG approach (1). Because no random bits are skipped during the congruential modulo-2 addition's output step, each iteration produces a one-bit random result. Because the linked LCG has the longest period, adding two connected-LCG outputs in the updated dual CLCG results in the same maximum period of 2n for a two-bit modulus operand. One XOR logic gate is all that is needed to execute modulo-2 addition. The memory footprint of a dual-large CLCG can be lowered while still completing a full 2-n period using the given PRBG methodology[7].

IV. RESULTS

Xilinx ISE is used to simulate and synthesise the suggested design.

16 BIT 3-OPERAND ADDER:

Simulation:

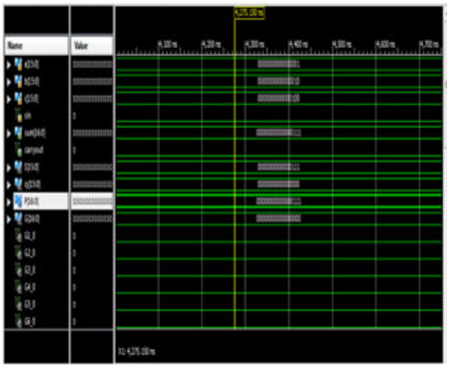


Fig3 . Simulation.

Synthesis Result:

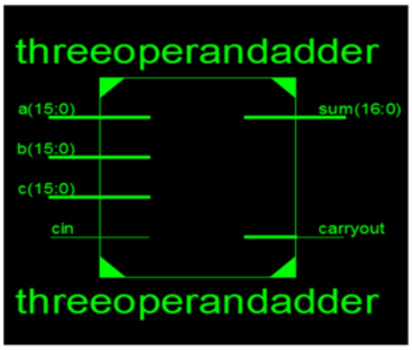
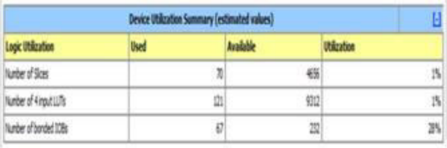
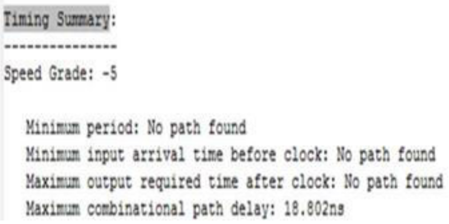


Fig.4 RTL Schematic.



Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	70	4656	1%
Number of 4 input LUTs	121	9312	1%
Number of bonded IOBs	67	232	28%

Fig.5 Design Summary.



```
Timing Summary:
---------------
Speed Grade: -5

   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: 18.802ns
```

Fig 6. Timing Summary.

32-BIT 3-OPERAND ADDER:

Simulation:

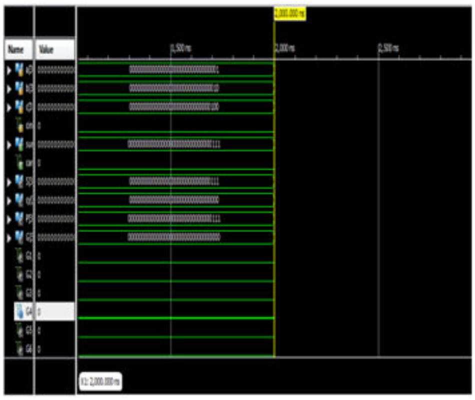


Fig 7. Simulation.

Synthesis Result:

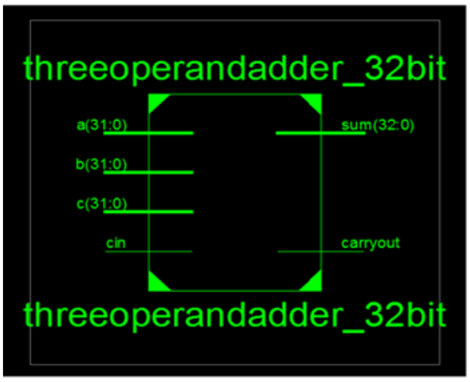
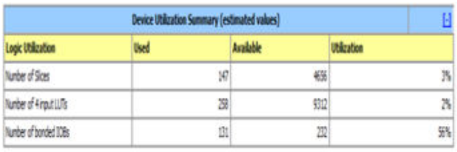
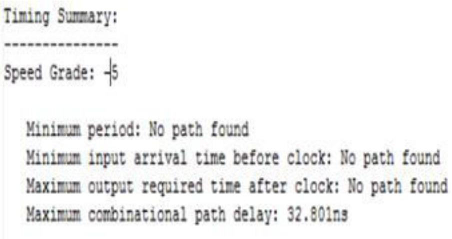


Fig 8. RTL Schematic.



Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	147	4656	3%
Number of 4 input LUTs	258	9312	2%
Number of bonded IOBs	131	232	56%

Fig.9 Design Summary.



```
Timing Summary:
---------------
Speed Grade: -5

   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: 32.801ns
```

Fig 10. Timing Summary.

MODIFIED DUAL CLCG RESULT:

Fig.11 Simulation.

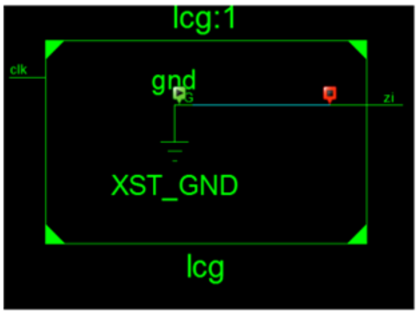


Fig12 . RTL Schematic.

V. CONCLUSION

This paper presents a three-operand binary addition approach and an efficient VLSI architecture for modular arithmetic computation in PRBG and cryptography applications. This design is well known for reducing critical route latency, area-delay product (ADP), and power-delay product (PDP) during the prefix calculation stages of PG logic and bit-addition logic. The design is verified on silicon chips using prototypes of the design on a commercially accessible FPGA platform and CS3A, the three-operand adder architecture proposed in this study.

REFERENCES :

[1]. Jiang, Y., Al-Sheraidah, A., Wang, Y., Sha, E., & Chung, J.-G. (2004). A Novel Multiplexer Based Low-Power Full Adder. IEEE Transactions on Circuits and Systems II: Express Briefs, 51(7), 345–348.

[2]. Bhattacharyya, P., Kundu, B., Ghosh, S., Kumar, V., & Dandapat, A. (2015). Performance Analysis of a LowPower High-Speed Hybrid 1-bit Full Adder Circuit. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 23(10), 2001–2008.

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

[4]. N.Umapathi, G.L. 2020. Design and Implementation of Low Power 16x16 Multiplier using Dadda Algorithm and Optimized Full Adder. International Journal of Advanced Science and Technology. 29, 3 (Feb. 2020), 918 - 926.

[5]. D. A. Patterson, J. L. Hennessy. "Computer Organization and Design: The Hardware/Software Interface, Fourth Edition". Morgan Kaufmann, Nov. 10, 2008.

[6]. F. Tenca. "Multi-operand Floating-Point Addition". In Proceedings of the 2009 19th IEEE Symposium on Computer Arithmetic (ARITH '09). IEEE Computer Society, Washington, DC, USA, pp.161-168, doi:10.1109/ARITH.2009.27.

[7]. Vishwa Shah, Urvisha Fata, Jagruti Makwana, " Design and Performance Analysis of 32 Bit VLSI Hybrid adder", Proceedings of the Third International Conference on Trends in Electronics and Informatics (ICOEI 2019) IEEE Xplore Part Number: CFP19J32-ART; ISBN: 978-1-5386- 9439-8.

[8]. R. Zimmermann, "Binary adder architectures for cell-based VLSI and their synthesis," Ph.D. thesis, Swiss Federal Institute of Technology, (ETH) Zurich, Zurich, Switzerland, 2019, Hartung-Gorre Verlag.

[9]. S.-R. Kuang, K.-Y. Wu, and R.-Y. Lu, "Low-cost high-performance VLSI architecture for montgomery modular multiplication," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 24, no. 2, pp. 434–443, Feb. 2016.

[10]. S.-R. Kuang, J.-P. Wang, K.-C. Chang, and H.-W. Hsu, "Energy-efficient high-throughput montgomery modular multipliers for RSA cryptosystems," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 21, no. 11, pp. 1999– 2009, Nov. 2013.

[11]. Srinivas, L., & Umapathi, N. (2022, May). New realization of low area and high-performance Wallace tree multipliers using booth recoding unit. In AIP Conference Proceedings (Vol. 2393, No. 1, p. 020221). AIP Publishing LLC.

[12]. N. Umapathi, G. M. Krishna and L. Srinivas, "A Comprehensive Survey on Distinctive Implementations of Carry Select Adder," 2021 4th Biennial International Conference on Nascent Technologies in Engineering (ICNTE), 2021, pp. 1-5, doi: 10.1109/ICNTE51185.2021.9487718.

[13]. Murali Krishna G., Karthick G., Umapathi N. (2021) Design of Dynamic Comparator for Low-Power and High-Speed Applications. In: Kumar A., Mozar S. (eds) ICCCE 2020. Lecture Notes in Electrical Engineering, vol 698. Springer, Singapore. https://doi.org/10.1007/978-981-15-7961-5_110

[14]. Swarnalatha, B., & Umapathi, N. (2022). Voltage over Scaling-Based Dadda Multipliers for Energy-Efficient Accuracy Design Exploration. Specialusis Ugdymas, 2(43), 2942-2956.

[15]. Prasad, R., UmapathI, N., & Karthick, G. (2022). Error-Tolerant Computing Using Booth Squarer Design and Analysis. Specialusis Ugdymas, 2(43), 2970-2985.

[16]. Saikrishna, D., Umapathi, N., & Mothe, S. (2022). Delays in the Generation of Test Patterns and in the Selection of Critical Paths. Specialusis Ugdymas, 2(43), 2986-2997.