



INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

HIGH SPEED AREA EFFICIENT VLSI ARCHITECTURE OF THREE OPERAND BINARY ADDER

G. Harshavardhan Reddy, K. Vaishnavi

Dept of Electronics & Communication Engineering,
TKR College of Engineering & Technology, India

DR. B. Swapna Rani

Associate Professor, Dept. of Electronics & Communication Engineering,
TKR College of Engineering & Technology, India

ABSTRACT

For Area-efficient VLSI architecture a Two-operand binary adder is designed where in which area is more and speed of operation is low, because of these limitations a new architecture with Three-operand binary adder logic is to be designed instead of previous architecture which performs 3-9 times faster than the previous architecture. Three-operand binary adder is the basic functional unit to perform the modular arithmetic and various Cryptography and Pseudorandom Bit Generator (PRBG) algorithm. Carry save adder (CS3A) is widely used technique to perform the Three-operand binary adder. However the ripple-carry stage in CS3A leads to a high propagation delay. Moreover a parallel prefix Two-operand binary adder such as Han-Carlson (HCA) can also be used for Three-operand addition that significantly reduces the critical path delay a cost of additional hardware. Three-operand binary addition significantly decreases area and power consumption. Hence a new High-speed and Area efficient adder architecture is proposed using pre-compute bitwise addition followed by carry prefix computation logic to perform the Three-operand binary addition. With this Logic a new architecture is to be designed, for 16-bit Three-operand binary addition which is called as 16-bit

Three-operand binary adder. To design this logic we are using Xilinx ISE, ModelSim software with VHDL, Verilog languages.

I. Introduction

Efficient and high-speed arithmetic circuits, such as binary adders, play a crucial role in digital system performance. Therefore, the development of VLSI architecture for a three-operand binary adder that is both area-efficient and fast is essential. A three-operand binary adder takes three binary numbers as inputs and generates their sum as output. The design of its VLSI architecture should aim to minimize the circuit's required area while maximizing its speed. Several techniques can be utilized, including pipelining, parallelism, and optimizing the circuit's layout. Advanced CMOS technology and circuit design methodologies like transistor sizing, gate sizing, and clock gating can also enhance the circuit's efficiency and performance.

In earlier, we used Ripple carry adder and Carry look ahead adder in being armature.

1. Ripple Carry Adder

Fig 1.2.1 refers to a Ripple Carry Adder which is a digital circuit that produces the computation sum of two double figures. A Ripple Carry Adder is constructed with the full adders connected in protruded w.ith the carry affair

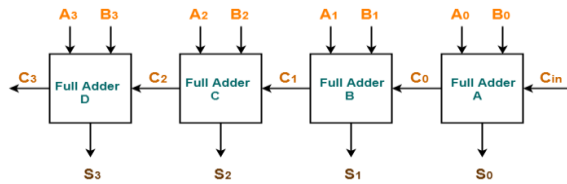


Fig 1: Ripple Carry Adder

From each full adder connected to the carry input of the coming full adder in the chain. Then 'n' number of ripple carry adders is used. Where the affair of one adder is given as input to coming adder i.e., coming adder should be awaited for the input which comes from former adder.

2. Carry Look Ahead Adder

A Carry Look Ahead Adder (CLA) are fast adder is a type of electronics adder is used in digital sense. Fig1.2 refers to Carry Look Ahead Adder. A Carry Look Ahead Adder improves the speed by reducing the quantum of time needed to determine the carry bits. But this CLA is only used for this lower number of bits; when we increase number of inputs the complexity also increases downsides OF Being Armature.

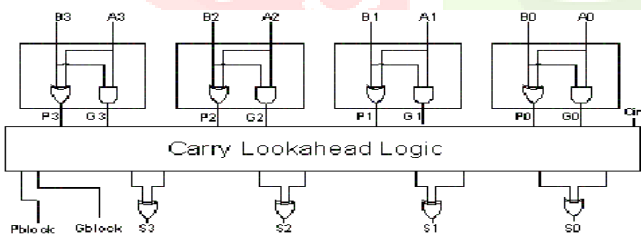


Fig 2: Carry Look Ahead Adder

By using the Ripple Carry Adder (RCA) we observed that one adder should stay for the input from the former adder by this there's further detention in the process. So to overcome this limitation we're going with the Carry Look Ahead Adder (CLA). A Carry Look Ahead Adder is used to reduce the detention means speed is increased compared to Ripple Carry Adder. But this Carry Look Ahead Adder is used for only lower number of bits, when we increase the number of bits automatically the sense gates(area) used are increased by this complexity also increases. To overcome the Ripple Carry Adder and

Carry Look Ahead Adder's downsides we're going with a new system i.e., proposed armature.

II. Proposed Architecture

The previous architecture utilized a ripple carry adder, which resulted in slow execution speeds and increased delay due to the dependence of each block or bit on the previous one. In order to enhance the speed and reduce the delay, a carry look ahead adder was introduced, resulting in faster execution times. However, the use of this adder is limited to a smaller number of bits, as increasing the number of bits can lead to a more complex process. A new architecture was proposed to address the limitations of the previous design, with the aim of improving speed and reducing delay and area. The project involves the use of Parallel Prefix Adders (PPA), specifically the Han-Carlson Adder (HCA), as well as Linear Congruential Generators (LCG) and Dual Linear Congruential Generator (DLCG). The current project employs Parallel Prefix Adders (PPA), including the Han-Carlson Adder (HCA), in addition to utilizing Linear Congruential Generators (LCG) and Dual Linear Congruential Generator (DLCG)

Parallel Prefix Adder

Multilevel-look Ahead adders or parallel-prefix adders can be utilized to overcome the delay of carry-look ahead adders. These adders operate by computing intermediate prefixes in small groups and then gradually combining them to determine the final carry bits. The structure of these adders is based on a tree-like architecture that resembles the carry propagate adder, with the addition of pre-computation and post-computation stages shown in the below fig 1.8. During the pre-computation stage, each bit performs a calculation to generate or propagate a carry, as well as to obtain a temporary sum. In the prefix stage, the carry generate/propagate signals of each group are computed to form a carry chain that provides carry-in for the adder below.

$$G_{i:k} = G_{i:j} + P_{i:j}$$

$$G_{j-1:k} P_{i:k} = P_{i:j} P_{j-1:k}$$

The final stage of the multilevel-look ahead adder or parallel-prefix adder is the post-computation stage, where the sum and carry-out are generated. However, if only the sum is required, the carry-out can be disregarded.

$$s_i = t_i \oplus G_{i:-1}$$

$$C_{out} = g_{n-1} + p_{n-1} _ G_{n-2:-1}$$

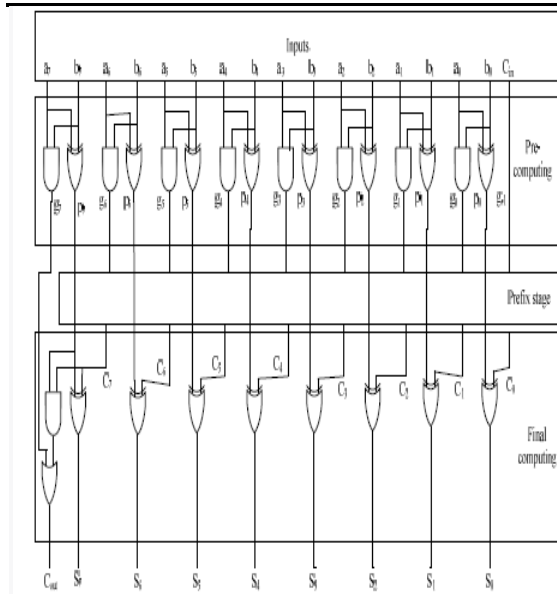


Fig 3:8-bit Parallel-Prefix Structure with carry save notation

Assuming that g_{i-1} is equal to c_{in} , where G_{i-1} is equal to c_i , the parallel-prefix structure diagram is presented in Figure 3.4, demonstrating an example with 8 bits. The equations mentioned earlier can be utilized to implement all parallel-prefix structures, but their interpretation can result in different types of trees, such as the Brent-Kung, which is recognized for its sparse topology but requires more logic levels. The performance of prefix structures can be influenced by several design factors, including radix/valency, logic levels, fan-out, and wire tracks. Structures, but their interpretation can result in different types of trees, such as the Brent-Kung, which is recognized for its sparse topology but requires more logic levels. The performance of prefix structures can be influenced by several design factors, including radix/valency, logic levels, fan-out, and wire tracks.

1. Han-Carlson Adder

Compared to other two-operand adder techniques, the Han-Carlson adder is known for its fast speed and low gate complexity, with the lowest area delay product (ADP) and power-delay product (PDP). As a result, the Han-Carlson adder (HCA) can be utilized to perform three-operand addition in two stages, as demonstrated in Figure 3.4.1.

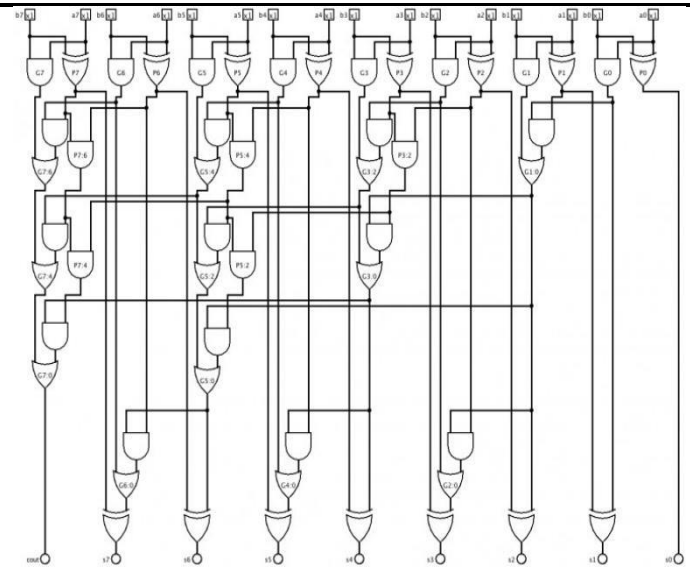


Fig 1.2 logic diagram of 8-bit Han-Carlson adder

The Han-Carlson prefix tree concept is comparable to Kogge-Stone's structure due to its high fan-out. However, the Han-Carlson prefix tree has the advantage of requiring significantly fewer cells and wire tracks than Kogge-Stone.

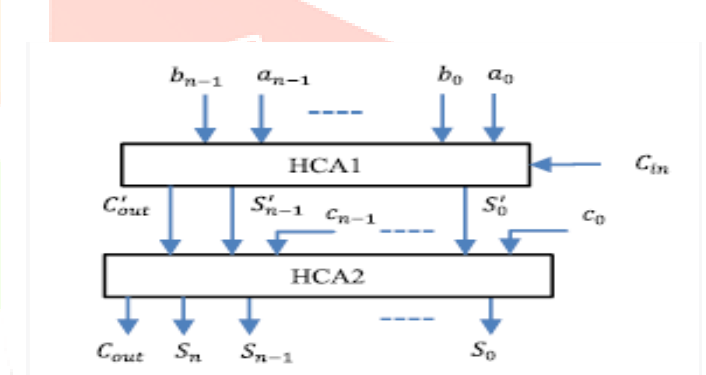


Fig 4: Block level architecture of HCA-based three-operand adder (HC3A)

3. Linear Congruential Generator (LCG)

An algorithm that produces a sequence of pseudo-random numbers using a piecewise linear equation is known as a linear congruential generator (LCG) as shown in the below figure 3.6.1.1. LCG is one of the most well-known and oldest pseudorandom number generator algorithms. These generators are easy to understand in terms of their underlying theory and are simple to implement. They are also fast, particularly on computer hardware capable of modular arithmetic via storage-bit truncation.

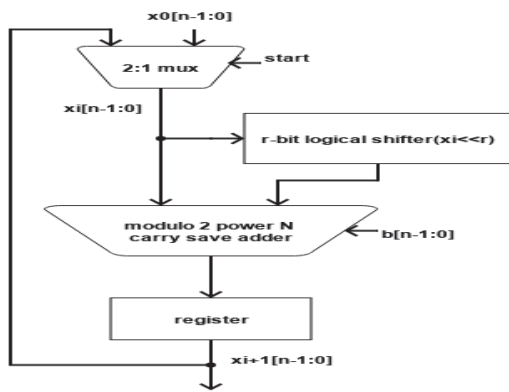


Fig 5: Architecture of the linear congruential generator

The design of the linear congruential generator involves the use of a 3-operand modulo $2n$ carry save adder, an n -bit 2×1 multiplexer, an n -bit register, and a logical shifter, as depicted in the figure. The LCG has low hardware complexity and occupies less area. However, due to its linear structure, it is not able to pass randomness tests since the next sequence can be identified by anyone after some time.

The architecture of the proposed "Modified Dual-CLCG" requires initial values of a_0, b_0, p_0, q_0 , four prime numbers $l_1, l_2, l_3, l_4 < 2n$, four numbers $m_1, m_2, m_3, m_4 < 2n$ such that $\prod_{i=1}^4 m_i - 1 \neq 0$, four LCG modules, and Magnitude Comparator (MC). Each LCG module comprises MUX, Register, R-generator, Shifter, Adder, and XOR gate. To implement the "Modified Dual-CLCG," the architecture requires several initial values such as a_0, b_0, p_0, q_0 , four prime numbers l_1, l_2, l_3 , and l_4 (less than $2n$), and four numbers m_1, m_2, m_3 , and m_4 (less than $2n$), such that $\prod_{i=1}^4 m_i - 1 \neq 0$. Additionally, the implementation requires four LCG modules, each consisting of a MUX, Register, R-generator, Shifter, Adder, and XOR gate, and a Magnitude Comparator (MC).

- MUX: The initial value, clock, and feedback of the LCG module are given as the inputs to the MUX. Initially, the value of feedback is zero.
- R-Generator: It is a switch that goes high only when the input satisfies, and is zero in other cases. Whenever the switch becomes high, the value of n is $2n-1$. For instance, when input is 5, then r is 2, i.e., $2 \times 5 - 1 = 5$.
- Shifter: It performs left shift operation r times on the a_i , where i is the round number. For example, if $r=2$, then a_i is shifted twice.
- Adder: It is designed by cascading n full adders in two rows.

- Register: This is used to store the value of Adder. The value stored in the register is the output of the LCG module, which is given as feedback.
- Magnitude Comparator: This is used to compare the values of two LCG modules based on the following criteria: If $a_i > b_i$ output=1 else output=0
- XOR gate: Performs XOR operation on the outputs of the two magnitude comparators. The output of the XOR gate is the output of the entire proposed system.

3.2 Operation

The system follows a set of steps, which begin by initializing a_0, b_0, p_0, q_0 to random numbers. These four values are given as one of the inputs to each mux, situated in lcg, respectively, while the other two inputs are clock and feedback. Initially, the clock is one, and feedback is zero. The mux's output is given to the r-generator, which decides the number of times the input value needs to be shifted. Later, the bits are shifted according to the value of the r-generator. The result of the shifting operation, the output of mux and 11 are given as inputs to an adder. The output of the Adder is stored in a register and is given as feedback to mux; it is also given as input to a magnitude comparator.

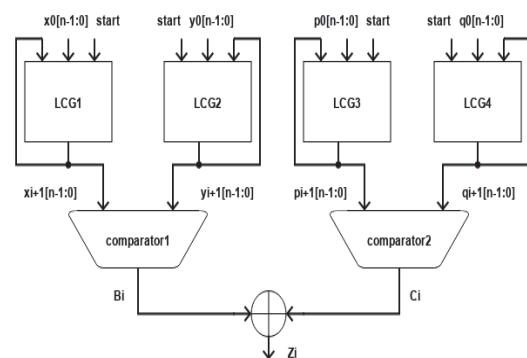


Fig 6: block diagram of dual linear congruential generator (DLCG)

The magnitude comparator takes two inputs, one from each lcg and then, compares every bit of the two inputs and gives a necessary output in the range of 0 to $2n$. For example: if the bit size is four, it generates 0 to 24, i.e., 0 to 15 binary numbers randomly. The primary advantage of a magnitude comparator is its inbuilt function which helps in generating a solution of a gate level equation without solving it.

Software used here is Vivado. Vivado is a software suite created by Xilinx, a leading semiconductor

company that provides programmable solutions for various industries. It is specifically designed to help engineers design and program Field-Programmable Gate Arrays (FPGAs) and Programmable System-on-Chip (SoC) devices. Vivado offers a comprehensive set of tools for design, simulation, implementation, and programming of these devices, making it a valuable tool for developers working on complex digital systems. With Vivado, engineers can achieve faster development cycles, higher performance, and greater productivity in their FPGA and SoC projects

III. Simulation Results

1. Han-Carlson Adder

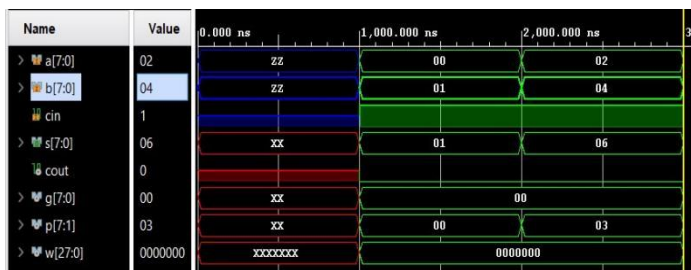


Fig 8: output waveform of Han-Carlson Adder

Based on the provided waveform, the system has several inputs, including a, b, and cin, as well as outputs p, g, w, s, and cout. Initially, values must be assigned to a, b, and cin. The adder itself consists of three stages: pre-computational, carry generation, and post-computational. The inputs a and b are first provided to the first stage, and the outputs of this stage are p and g. These p and g values are then passed as inputs to the second stage, which produces the output w. The value of w is then given as input to the third and final stage, which outputs s and cout.

Power Report

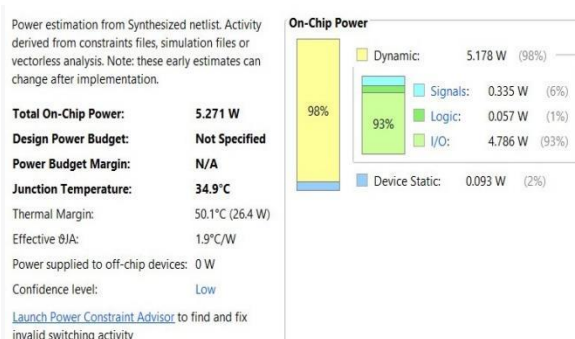


Fig 9: power report of Han-Carlson Adder

The power report of a Han-Carlson adder shows the amount of power consumed during operation. It can be generated using simulation software and includes information such as total power consumption, dynamic power, and leakage power. The Han-Carlson adder is designed to reduce power

consumption by minimizing logic gates and using efficient gate configurations. Analyzing the power report can help optimize the circuit's design to minimize power consumption while maintaining performance.

Utilization Report

A utilization report for the Han-Carlson Adder analyzes the circuit's resource usage, including logic cells, registers, and routing resources. This report helps to optimize resource usage, improve performance, and reduce power consumption. It is generated by design software and provides detailed information on the percentage of resources used and the number of logic cells, registers, and routing resources used. By analyzing the report, designers can optimize routing, reduce the number of logic cells, and use efficient gate configurations to improve efficiency and meet design requirements.

Name	Slice LUTs (41000)	Slice (10250)	LUT as Logic (41000)	Bonded IOB (300)
hancarlson	11	5	11	24

Fig 10: Utilization report of Han-Carlson Adder

2. Linear Congruential Generator (LCG)

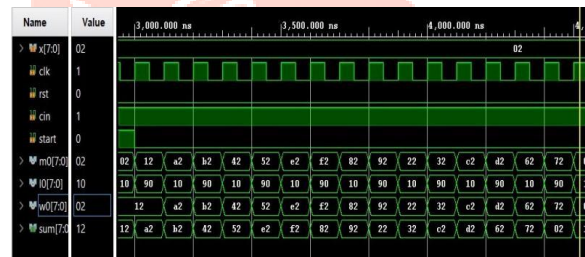


Fig 12: output waveform of LCG

Based on the provided waveform, the system has several inputs including X0, clock, reset, start, and cin, as well as outputs m0, i0, and sum. Initially, a random value is assigned to X0. Subsequently, the appropriate signals must be applied to clock, reset, start, and cin as necessary. Once X0 is applied, it is passed through a multiplexer (MUX), which outputs m0. The value of m0 is then shifted using a shifter that shifts 3 bits, producing i0 as an output. Both m0 and i0 are then inputted into an adder, and the resulting sum is stored in a register. This sum is then outputted as the final value.

Power Report

Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power: 5.205 W
Design Power Budget: Not Specified
Power Budget Margin: N/A
Junction Temperature: 34.8°C
 Thermal Margin: 50.2°C (26.5 W)
 Effective θ JA: 1.9°C/W
 Power supplied to off-chip devices: 0 W
 Confidence level: Low
[Launch Power Constraint Advisor](#) to find and fix invalid switching activity

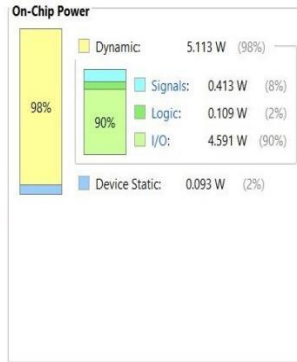


Fig 13: Power report of LCG

The power consumption of a linear congruential generator is affected by the complexity of the multiplier and the clock frequency. Low-power components and clock gating techniques can be used to reduce power consumption. Increasing the size of the shift register also increases power consumption.

Utilization Report

The linear congruential generator is widely used in various applications that require pseudorandom numbers, such as cryptography, simulations, and games. In cryptography, it is used for key generation and data encryption. In simulations, it is used to create realistic scenarios in various fields such as finance, engineering, and social sciences. In gaming, it is used for game mechanics and procedural content generation. Additionally, linear congruential generators can be used in statistical sampling and Monte Carlo simulations to estimate the probability of complex events.

Name	Slice LUTs (41000)	Slice Registers (82000)	Slice (10250)	LUT as Logic (41000)	Bonded IOB (300)	BUFGCTRL (32)
lcg_ppa	10	8	3	10	19	1
u3 (dff_4)	10	8	3	10	0	0

Fig 14: Utilization Report of LCG

3. Dual Linear Congruential Generator (DLCG)

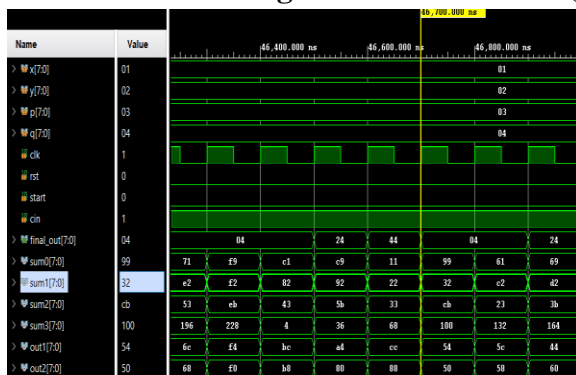


Fig 16: Output Waveform of DLCG

Based on the waveform, the system has several inputs, including x0, y0, p0, q0, and start, and outputs, including sum0, sum1, sum2, sum3, out1, out2, and the final output. Initially, we assigned the values 1, 2, 3, and 4 to the inputs x0, y0, p0, and q0, respectively, which serve as inputs to four linear congruential generators (LCGs). The outputs of these LCGs are sum0, sum1, sum2, and sum3. The values of sum0 and sum1 are then inputted into comparator1, while sum2 and sum3 are inputted into comparator2. The outputs of these two comparators are out1 and out2. Finally, the final output is generated through the XOR operation using out1 and out2 as inputs.

Power Report

Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power: 8.427 W
Design Power Budget: Not Specified
Power Budget Margin: N/A
Junction Temperature: 40.9°C
 Thermal Margin: 44.1°C (23.3 W)
 Effective θ JA: 1.9°C/W
 Power supplied to off-chip devices: 0 W
 Confidence level: Low
[Launch Power Constraint Advisor](#) to find and fix invalid switching activity

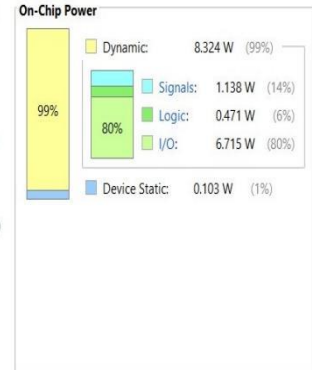


Fig 17: Power Report of DLCG

The power consumption of a dual linear congruential generator depends on the implementation of its components and clock frequency. The use of two generators and an XOR gate increases the overall complexity of the circuit, resulting in higher power consumption. Low-power components, clock gating techniques, and optimized shift register size can be used to reduce power consumption.

Utilization Report

The dual linear congruential generator is a popular choice for applications requiring high-quality pseudorandom numbers due to its ability to provide a larger range of values and a more uniform distribution compared to a single generator. It is easy to implement and can be optimized for power consumption, making it a versatile option for various applications.

Name	Slice LUTs (41000)	Slice Registers (82000)	Slice (10250)	LUT as Logic (41000)	Bonded IOB (300)	BUFGCTRL (32)
dual_lcg_ppa	52	32	15	52	43	1
u0 (lcg_ppa)	19	8	9	19	0	0
u1 (lcg_ppa_0)	11	8	3	11	0	0
u2 (lcg_ppa_1)	11	8	3	11	0	0
u3 (lcg_ppa_2)	11	8	5	11	0	0

Fig 18: Utilization Report of DLCG

CONCLUSION

In conclusion, high-speed, area-efficient VLSI architecture for a three-operand binary adder is a promising development in the field of digital circuit design. This architecture offers advantages over traditional binary adders, including higher speed, greater area efficiency, lower power consumption, flexibility, and accuracy. The applications of this architecture are numerous, ranging from image and video processing to automotive and aerospace systems, financial applications, gaming, and the Internet of Things (IoT).

FUTURE SCOPE

The future scope of this architecture is wide-ranging, with potential applications in emerging fields such as quantum computing and AI, as well as established fields such as 5G and edge computing. Ongoing research and development in the field are likely to lead to new and innovative applications of this architecture in the future. Overall, high-speed, area-efficient VLSI architecture for a three-operand binary adder is a valuable development in the field of digital circuit design, with the potential to improve the performance and efficiency of a wide range of applications.

REFERENCES

- [1] A. K. PANDA and K. C.RAY "*Modified Dual-CLCG Method and its VLSI Architecture for Pseudorandom Bit Generation,*" in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 66, no. 3, pp. 989-1002, March 2019.
- [2] A.KUMAR PANDEY, K.CHANDRA RAY and RAKESH PALISHETTY "*High Speed and Area Efficient VLSI architecture*" IEEE, NOV, 2020.
- [3] K.SAMBHAV PANDEY and NEERAJ GOEL "*Ultra Fast Parallel Prefix Adders*" IEEE, JUNE, 2019.
- [4] S.M SUDHAKAR and K. P. CHIDAMBARAM "*Hybrid Han-Carlson adder*" IEEE, AUG 2012.
- [5] A.KUMAR PANDEY and K.CHANDRA RAY "*Modified Dual-CLCG Method and its VLSI architecture of Pseudo Random Bit Generator*" IEEE, MARCH 2019.
- [6] R. S. KATTI and R. G. KAVASSERI, "*Secure pseudo-random bit sequence generation using coupled linear congruential generators,*" 2008 IEEE International Symposium on Circuits and Systems, 2008, pp. 2929-2932.

[7] PREET, RAMINDER ET AL. "*Performance Analysis of 32-Bit Array Multiplier with a Carry save Adder and with a Carry-Look- Ahead Adder.*" (2009).

[8] P.PERIS-LOPEZ, E.SANMILLIAN, J.C.A.VAN DER LUBBE and L.A.ENTERNA, "*Cryptographically secure pseudo-random bit generator for RFID tags,*" 2010 pp. 1-6.

[9] KAUR, JASBIR and L.SOOD, "*Comparison between Various Types of Adder Topologies.*" (2015).

[10] R. S. KATTI and S. K. SRINIVASAN, "*Efficient hardware implementation of a new pseudo-random bit sequence generator,*" 2009

IEEE International Symposium on Circuits and Systems, 2009, pp. 1393- 1396.

