International Journal of Scientific Research in Science and Technology
Print ISSN: 2395-6011 | Online ISSN: 2395-602X (www.ijsrst.com)
doi : https://doi.org/10.32628/IJSRST

Design and Synthesis of Multi-Operand Parallel Prefix Adder for Pseudorandom Bit Generator Applications

Korrapati Vyasa Purnima [1], K.Mohana Krishna [2]

Student[1], Dept of ECE, Shree Institute of Technical Education, Tirupati, Andhra Pradesh, India.

Asst.Professor[2], Dept of ECE, Shree Institute of Technical Education, Tirupati, Andhra Pradesh, India.

Article Info

Volume 9, Issue 5
Page Number : 493-498

Publication Issue

September-October-2022

Article History

Accepted : 10 Oct 2022
Published : 24 Oct 2022

ABSTRACT

Numerous cryptographies as well as pseudorandom bit generator (PRBG) algorithms are employing the 3-operand binary adder as the basic functional block to accomplish modular arithmetic. It is also used in a variety of other purposes. To realize the three-operand binary addition, the current technique includes a high-speed as well as area-efficient adder design related to pre-bit -wise addition with carry prefix processing mechanism. In relation to previous suggested approach such as the 3 operands carry save adder and the 2 operands oriented three operand HCA. Rather than the Han-Carlson adder, we were employing the Ladner Fischer adder. Because this takes up less space and has a shorter latency. The synthesis and simulation are verified by employing Xilinx ISE 14.7 Tool.

Keywords: Binary Adder, Parallel prefix adders, Pseudorandom bit generators, Cryptography.

I. INTRODUCTION

Implementing cryptographic techniques on hardware is required to provide optimal system performance by ensuring physical security. Multiple cryptographic techniques employ modular arithmetic for arithmetic operations which including modular exponentiation, modular multiplying, as well as modular adding. The effectiveness of the cryptography system will be directly affected by the congruential modular arithmetic process.

If the bit is beyond 32 bits, then it is polynomial-time unexpected but safe. Like a result, as any operand value increases in size, its MDCLCG's protection increases. Because its hardware architecture comprises of 4 three-operand modulo-2n adders, 2 comparators, with 4 mux in research previously, the area as well as critical path time effect increases. As a result, by effectively implementing the three-operand adder, the MDCLCG's efficiency can be enhanced.

Two operand Adders

Two operand adders are adders that execute adding among two operands or else two n-bit values provided as input source. To conduct such actions, adders including such ripple carry full adder, parallel prefix adder circuit, carry skip full adder, and etc are often used.

Ripple Carry Adder

The RCA is the primary 'n' bit FA design. It allows for the addition of two operands. The RCA is reported to be made up with cascading of Full Adders.

An outcome under each step is acquired in a ripple-like pattern, with carry transmitting towards each level. Because the RCA is done purely with FAs, the area overhead is minimal. However, the greatest disadvantage is the time it takes to provide the results. Thus, every phase is reliant mostly on preceding one's carry.

Parallel Prefix Adders

Employing parallel prefix full adder, the basic objective is to calculate a tiny number of intermediary prefixes, whereupon locate next huge mass prefixes, and so forth, till every carry values were determined. So addition to creating the ultimate sum function, parallel prefix full - adder can be divided into three phases: 1. Initial-processing. 2. Carry signal creation 3. Ultimate Executing

Three operand Adder:

Trio operand full adder were adders that execute adding among 3 variable or else three input n-bit values. This CSA is a 3-operand adder that works with n-bit values.

Carry save Adder:

The carry save full adder is an electronics full adder which can rapidly combine 3 or even more digital sets of data. It varies from some of the other electronic full adder because it also produces two (or more) digits, which can be added together will produce the result to the initial total. Because a binary multiplier includes the adding of more than two binary integers following multiplication, a carry saving adder is commonly employed in binary multipliers.

Two steps are involved in the Carry save addition. The collection of full adders is the very first step, and these adders have no connectivity. So, every full adder calculates and saves carry as well as sum bits over the next phase. To obtain the overall summation output, the step 2 uses a ripple carry adder.

Two different two-operand full adder or perhaps 3 operand full adder that are employed to execute the 3-operand binary addition. The carry-save 3-operand adder is a space-efficient as well as commonly employed approach for performing three-operand binary combination employing modular arithmetic, which is utilized within cryptography techniques to ensure security.

The remaining portion of this work is laid out as follows: The three-operand adder using Han Carlson adder architecture is introduced in Section II. The suggested 3-operand adder architecture using Ladner Fischer adder is thoroughly described in Section III. Section IV contains the experimental outcomes. Section V deals with conclusion.

II. EARLIER WORK

The most common adder technology is a parallel prefix adder. The prefix adder, on either side, uses four-stage procedures rather than three to assess the sum of three binary values operands, comprising bit-adding logic, base logic, PG logic, finally summing logic. All four steps have the following logical expression:

Phase-1: Bit Addition Logic:

$$S'_i = a_i \oplus b_i \oplus c_i$$

$$cy_i = a_i \,.\, b_i + b_i \,.\, c_i + c_i \,.\, a_i$$

Phase-2: Base Logic

$$G_{i:j} = G_i = S'_i \, cy_{i-1}, \qquad G_{0:0} = G_0 = S'_0 \, C_{in}$$

$$P_{i:j} = P_i = S'_i \oplus cy_{i-1}, \qquad P_{0:0} = P_0 = S'_0 \oplus C_{in}$$

Phase-3: PG (Propagate & Generate)Logic

$$G_{i:j} = G_{i:k} + P_{i:k} \,.\, G_{k-1:j}$$

$$P_{i:j} = P_{i:k} \,.\, P_{k-1:j}$$

Phase-4: Sum Logic

$$S_i = (P_i \oplus G_{i-1:0}), \quad S_0 = P_0, \quad C_{out} = G_{n:0}$$

The 3-operand binary adder's conventional VLSI layout as well as internal workings are represented in the diagram following.

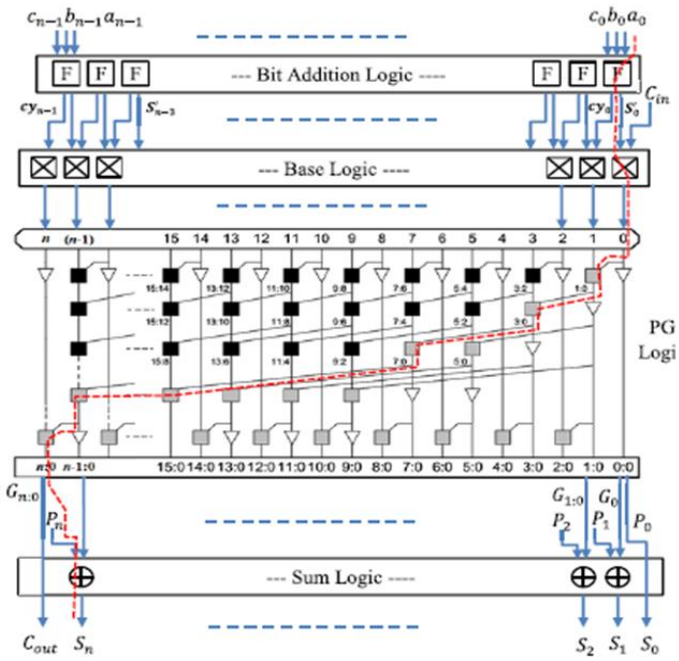


Fig1: Existing three operand adder

During first step, the typical full adder's desired output sum (Si) bit and also the right-adjacent full adder's output carry bit were combined to calculate the generate Gi as well as propagate (Pi) signs (base logic). The squared saltire-cell, as can be seen in Figure.2, is used to reflect the calculating of Gi as well as Pi values, and the base logic phase contains n+1 saltire-cells.

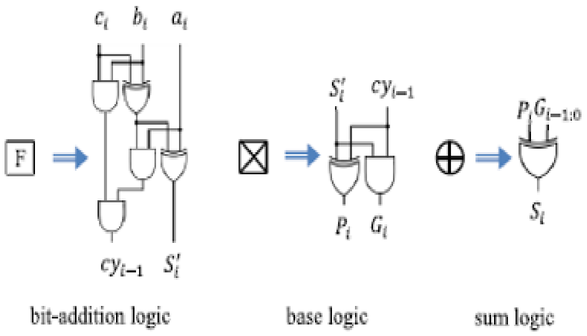


Fig2. Logical diagram of bit addition, base logic, sum logic

The carry estimation level, also known as generate as well as propagate logic (PG), the carry bit is pre-computed using a combination of black & grey cell logics. Figure 3 depicts a black & grey cell's logical structure which evaluates the carry create Gi: j as well as propagate Pi: j

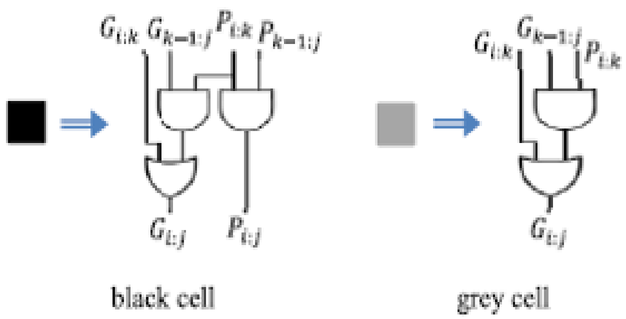


Fig3. Logic diagram black-cell and grey-cell.

This carry propagate sequence has a considerable impact on the present adder's critical path duration because the recommended adder has (log2 n+1) prefix calculation steps. In the last phase, which is symbolized as summing logic. The carryout response (Cout) can be stated that an individual as from carry generate value.Gn:0.

Han-Carlson adder

Among Brent-Kung and Kogge-Stone trees, Han Carlson trees belong to a family of tree structures. This tree uses Kogge-Stone upon that odd-numbered bits before employing another phase to ripple seen between even places. There are three main processes in total: pre-processing, carry generation, and the last one is post-processing. The propagating as well as generate data are found during the pre-processing phase, as well as the summation parameters are estimated during the post-processing phase, as illustrated in the diagram. For each parallel prefix adder, the carry generation is different. Here is a diagram of the Han-Carlson adder carry generation.

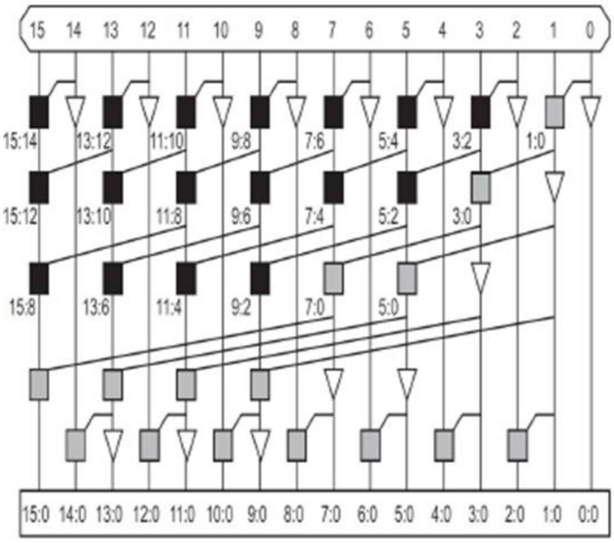


Fig4. The carry generation structure of Han-Carlson adders

Leveraging dual operand adders, a 3-operand adder could be created. Dual operand parallel prefix full adder are still used to speed up the process of the operation. Whenever the bit size grows, the Han-Carlson parallel prefix adder is the quickest of all. 2 HCA adders are required to complete the 3-operand adding.

In relative terms to the 3-operand CSA, the HC3A significantly reduces critical route latency. In addition, as the bit size of the adder grows, so does the area of the adder. As a conclusion, in created models, a modern high-speed as well as area-efficient 3-operand adder technique is combined with an advantageous VLSI architecture to lessen the area-delay trade-off.

III. PROPOSED WORK

This section employs the Ladner Fischer Adder to construct a new three-operand binary adder construction. In HCA the need of black cells and grey cells is more. Therefore, the area complexity is high in Han Carlson Adder. In order to overcome that, here proposing Ladner Fischer Adder.

FPGAs have become extremely popular in present generation due to their ability to enhance the speed of microprocessor related applications such as mobile communication, DSP, as well as telecommunication. The creation of gadgets is derived from research on binary operations as well as motivation.

The proposed solution uses the parallel prefix Ladner Fischer adder, as well as its VLSI structure, to perform three operands adding in modular arithmetic. Parallel prefix adders are the most prevalent adder technology. The prefix adder, on either side, uses four-stage procedures rather than three to assess the sum of three binary values operands, comprising bit-adding logic, base logic, PG logic, finally summing logic.

Ladner Fischer adder:

The prefix tree of the Ladner Fischer adder is basic. The Sklansky is a superstructure that divides as well as overcomes. By combining two smaller full adders on every stage, this Arrangement iteratively evaluates prefixes for 2-bit segments, 4-bit communities, 8-bit communities, 16-bit communities, and etc.

Carry generation phase Ladner Fischer adder: It calculates intermediary prefix bits together with large community prefixes to offer the latency of log2n phases employing the split as well as overcome method. The benefit of this adder is that its construction is simple as well as consistent, but it still has a fan-out issue because fan-outs double at every level.

Maximum fan-out: (n/2) +1.

The propagate as well as generate data are found during the pre-processing phase, and also the aggregate parameters are estimated during the post-processing phase, as illustrated in the diagram. To every parallel prefix adder, the carry generation is different. Here is a diagram of the Ladner Fischer adder carry production.

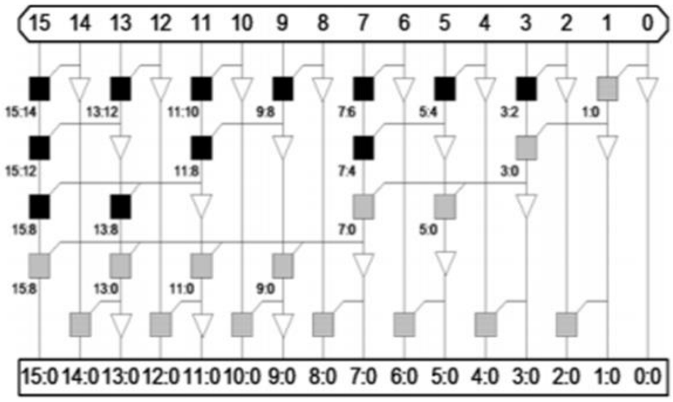
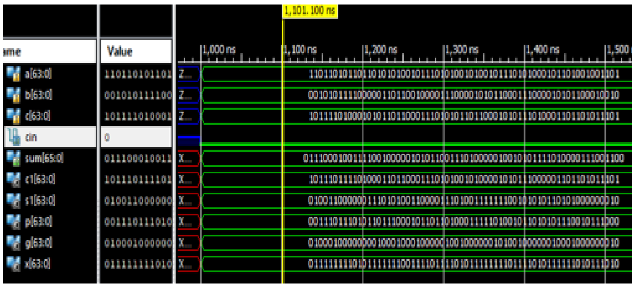


Fig4: Ladner Fischer Carry Diagram

Leveraging 2 operand full adder, a 3-operand adder can be created. Dual operand parallel prefix full adder are often used to increase the frequency of the operations. Whenever the bit size grows, the Ladner Fischer is the quickest across all types of parallel prefix adders. Employing 2 Ladner Fischer adders, a 3-operand accumulation can be done.

IV. EXPERIMENTAL RESULTS

In this paper, the proposed Ladner Fischer Adder architectures all are implemented using Verilog HDL with Xilinx ISE 14.7 tool.



```
ame        Value
a[63:0]    110110101101   Z...  1101101011011010101001011101010010100101110101000101101001001101
b[63:0]    001010111100   Z...  0010101111000001101100100001110000101011000111000010101100010010
c[63:0]    101111010001   Z...  1011111010000101011011000111010101101100010101110100011011010111101
cin        0
sum[65:0]  011100010011   X...  011100010011110010000010101100111010000010010101111010000111001100
c1[63:0]   101110111101   X...  1011101111100001101100011101010010100001010111000001101101011101
s1[63:0]   010011000000   X...  0100110000001110101001100001110100111111100101010110101000000010
p[63:0]    001110111010   X...  0011101110101101110001011010100011111010010110101011100101111000
g[63:0]    010001000000   X...  0100010000000010001000100000100100000001010010000001000100000010
x[63:0]    011111111010   X...  0111111110101111111100111101111010111111111011110101111110101110
```

Fig7: Simulation results of proposed Ladner Fischer-based three operand adder

Figure 7 shows the simulation results for the three-operand adder using Ladner Fischer Carry production.

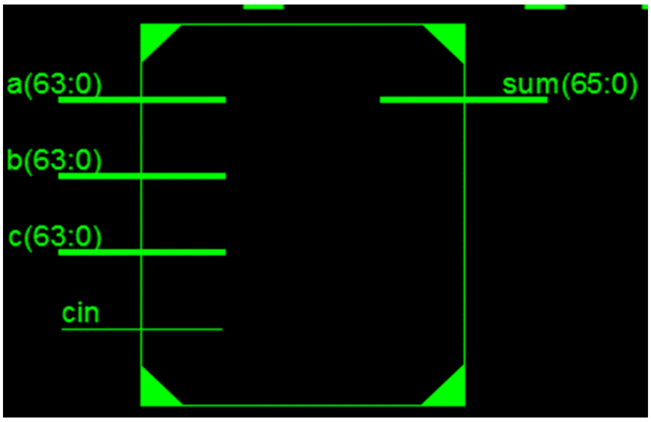


Fig 8: Block diagram of proposed 3-operand adder

Figure 8 shows the block diagram of three-operand adder using Ladner Fischer carry production architecture.

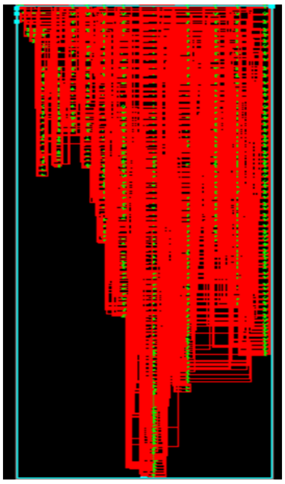


Fig 9: Technology schematic of proposed 3-operand adder

Figure 9 shows the technology schematic of three-operand adder using Ladner Fischer carry production architecture

Comparison between existing method and proposed method

	Area	Delay
Existing(Han-Carlson)	368	7.531ns
Proposed(Ladner-Fischer)	246	20.068ns

Table 1: comparison between Existing and Proposed methods.

Table 1 describes the comparison of performance parameters such as Delay and Area between Existing and Proposed for three operand adder designs. From these results, we conclude that the area is reduced in proposed method by using Ladner Fischer carry production.

V. CONCLUSION

The 3-operand binary summation is carried out in different cryptographic techniques. The suggested architecture is unique as it also reduces size in the prefix calculation phases of PG logic as well as bit-addition processing, leading to decrease in critical path latency altogether. When compared to certain other parallel prefix 3 operand full adder throughout the previous approaches, such as the Han-Carlson Adder, also it reduces the area overhead. Utilizing Xilinx ISE software, the synthesis as well as simulation are confirmed.

VI. REFERENCES

[1]. 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.

[2]. Z. Liu, J. GroBschadl, Z. Hu, K. Jarvinen, H.Wang, and I. Verbauwhede, 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.

[3]. 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.

[4]. B. Parhami, Computer Arithmetic: Algorithms and Hardware Design. New York, NY, USA: Oxford Univ. Press, 2000.

[5]. P. L. Montgomery, Modular multiplication without trial division, Math. Comput., vol. 44, no. 170, pp. 519–521, Apr. 1985.

[6]. 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.

Cite this article as :

Korrapati Vyasa Purnima, K. Mohana Krishna, "Design and Synthesis of Multi-Operand Parallel Prefix Adder for Pseudorandom Bit Generator Applications", International Journal of Scientific Research in Science and Technology (IJSRST), Online ISSN : 2395-602X, Print ISSN : 2395-6011, Volume 9 Issue 5, pp. 493-498, September-October 2022.

Journal URL : https://ijsrst.com/IJSRST229591