ISSN 2714-7533

FPGA Implementation of High Speed and Area Efficient Three Operand Binary Adder

Saba Azeez [a,1], Pankaj Rangaree [a,2,*]

[a] E & C Department Vaagdevi College Warangal, India
[1] sabaazeez123@gmail.com; [2] phrangaree247@gmail.com
* corresponding author

ARTICLE INFO

Article history
Received February 12, 2021
Revised March 10, 2021
Accepted April 13, 2021

Keywords
three operand adder
Han-Carlson adder
MDCLCG
pre-compute bit-wise addition
carry prefix computation logic

ABSTRACT

Three operand binary adder is the basic functional unit to perform the pseudorandom bit generator algorithms and in various cryptography. The basic method used to perform the three-operand binary addition is carry save adder, which leads to high delay. For this a parallel prefix two operand adder such as Han-Carlson adder is used to reduce the delay but increases the hardware architecture i.e., area increases. To overcome this disadvantage, we need a new area efficient and high-speed adder architecture to be proposed using pre compute bitwise addition followed by carry prefix computation logic to perform three operand binary adder which reduces delay and area efficiently. This method is the proposed method and implemented on the FPGA device. A newly designed three operand binary adder is shown and is implemented in MDCLCG. The results of 16 bit and 32-bit three operand adder will be shown and this proposed method is applied on Modified Dual CLCG. The Carry-Save-Adder architecture used in 32-bit MDCLCG is replaced by the proposed architecture. The design is prototyped on a commercially available FPGA platform to validate the design on silicon chip.

1. Introduction

For good optimal system performance while maintaining physical security its necessary to organise the cryptographic algorithms on hardware. The performance of the cryptography algorithms depends on modular arithmetic operation. Montgomery algorithm is an important part of modular arithmetic operation whose critical operation is based on three operand binary addition. The primary arithmetic operation in LCG (Linear Congruential Generator) based pseudo random generators is three operand binary adder. Among Coupled LCG, Modified Dual CLCG, Coupled Variable Input LCG- Modified Dual CLCG is the most secure and random pseudo random bit generator method and its area and delay increases with increase of operand size. The performance of MDCLCG can be improved by using three operand binary adder. The basic method used in three operand binary adder is carry save adder but due to ripple carry adder stage in carry save adder it gives a larger delay. To overcome this, we are using a parallel prefix adder that is Han-Carlson Adder which reduces the delay but increases area by increasing the hardware architecture. To have a fast three operand binary addition operation we need an efficient VLSI architecture which reduces both the delay and the area. For this we are using pre compute bit-wise addition followed by the carry prefix computation logic which reduces area and delay efficiently (proposed method).

The Objective of the project is to reduce area and delay by using the proposed method in three operand binary addition. This proposed method is then applied to Modified dual CLCG. We are showing 16-bit and 32-bit three operand binary adder by using proposed method and also applying this method in 32-bit MDCLCG.

The paper is organised as follows section II- Literature Survey, section III- Proposed method and MDCLCG with proposed method, section IV- Results, section V-Conclusion.

2. Literature Review

The necessity of hardware security for internet-of-things applications demands a low hardware area, high speed and secure pseudorandom bit generator (PRBG). Amongst various PRBGs, Blum-Blum-Shub (BBS) is the proven cryptographically secure PRBG because of its large prime factorize problem. The efficient implementation of BBS method relies on the large integer modular multiplication which makes it computationally expensive. Montgomery algorithm is a very efficient solution to perform the modular multiplication which replaces the critical trial division with series of shift and additions. However, the clock latency and critical path delay are increased with increase of modular size. Therefore, in this paper, a modified radix-2 iterative Montgomery modular multiplier is used for efficient hardware implementation of 1024-bit BBS generator. It replaces two two-operand adders with one three-operand adder. Carry-save adder is the commonly used technique for three-operand addition which experiences high critical path delay. Hence, the critical path delay is further reduced by employing a fast parallel prefix Han-Carlson adder for three-operand addition in the proposed architecture. The proposed architecture is designed using Verilog HDL and prototyped on the Virtex5 FPGA device. The physical implementation results report that the proposed 1024-bit BBS architecture can work at a maximum frequency of 71.2 MHz with overall latency improvement of 93.87%.

Pseudorandom bit generator (PRBG) is an essential component for securing data during transmission and storage in various cryptography applications. Among popular existing PRBG methods such as linear feedback shift register (LFSR), linear congruential generator (LCG), coupled LCG (CLCG), and dual-coupled LCG (dual-CLCG), the latter proves to be more secure. This method relies on the inequality comparisons that lead to generating pseudorandom bit at a non-uniform time interval. Hence, a new architecture of the existing dual CLCG method is developed that generates pseudo-random bit at uniform clock rate. However, this architecture experiences several drawbacks such as excessive memory usage and high-initial clock latency, and fails to achieve the maximum length sequence. Therefore, a new PRBG method called as “modified dual-CLCG” and its very large-scale integration (VLSI) architecture are proposed in this paper to mitigate the aforesaid problems. The novel contribution of the proposed PRBG method is to generate pseudorandom bit at uniform clock rate with one initial clock delay and minimum hardware complexity. Moreover, the proposed PRBG method passes all the 15 benchmark tests of NIST standard and achieves the maximal period of 2^n. The proposed architecture is implemented using Verilog-HDL and prototyped on the commercially available FPGA device.

3. Proposed Method

3.1. Three Operand Binary Adder

Three operand binary adder is the basic functional unit to perform the pseudorandom bit generator algorithms and in various cryptography. For good optimal system performance while maintaining physical security its necessary to organise the cryptography algorithms on hardware. The performance of the cryptography algorithms depends on modular arithmetic operation. Montgomery

algorithm is an important part of modular arithmetic operation whose critical operation is based on three operand binary addition. The primary arithmetic operation in Linear Congruential generator (LCG) based pseudo random bit generators such as Coupled LCG, Modified Dual CLCG and Coupled Variable Input LCG are the three-operand binary addition. Among this LCG based PRBG methods Modified Dual CLCG is the most secure and highly random pseudo random bit generator method. Its area and delay increase with increase in operand size. The delay and area of Modified Dual CLCG can be improved by the implementation of efficient VLSI architecture for three operand binary adder.

The basic method used in three operand binary adder is carry save adder but the ripple carry stage in carry save adder leads to high delay so to overcome this we can use a parallel prefix adder which is Han-Carlson adder. It reduces the delay efficiently but increases the hardware architecture i.e., area. In recent years, various such kind of parallel prefix two-operand adders. The ultra-fast adder is reported as the fastest one, and it is even faster than the Han-Carlson by three gates delay. However, it consumes comparatively two times large gate area than the Han-Carlson adder. Therefore, to minimize this trade-off between area and delay, a new high-speed, area-efficient three-operand adder technique and its efficient VLSI architecture is proposed in the next section. To reduce the area and also the delay we should implement an efficient VLSI architecture i.e., a pre compute bitwise addition followed by the carry prefix computation logic which reduces the parameters significantly. This pre compute bit wise addition followed by carry prefix computation logic is the efficient method to be used in three operand binary addition which reduces both delay and area than other adders. The proposed adder technique is a parallel prefix adder. However, it has four stage structures instead three-stage structures in prefix adder to compute the addition of three binary input operands such as bit-addition logic, base logic, PG (propagate and generate) logic and sum logic.

The new adder technique performs the addition of three n-bit binary inputs in four different stages. In the first stage (bit-addition logic), the bitwise addition of three n-bit binary input operands is performed with the array of full adders, and each full adder computes “sum (Si)” and “carry (ci)” signals as highlighted in. The logical expressions for computing sum (Si) In the first stage, the output signal “sum (Si)” bit of current full adder and the output signal “carry” bit of its right-adjacent full adder are used together to compute the generate (Gi) and propagate (Pi) signals in the second stage (base logic). The computation of Gi and Pi signals are represented by the “squared saltire-cell” and there are n number of saltires `cells in the base logic stage.

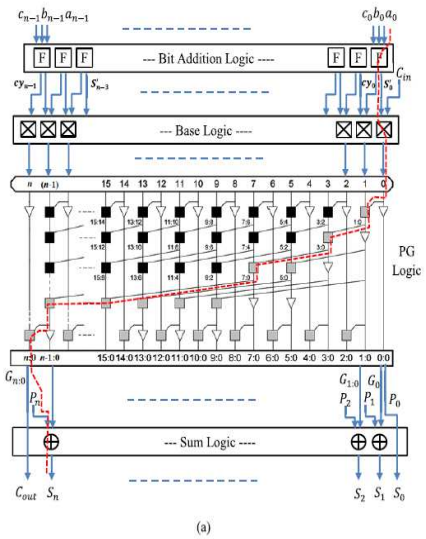
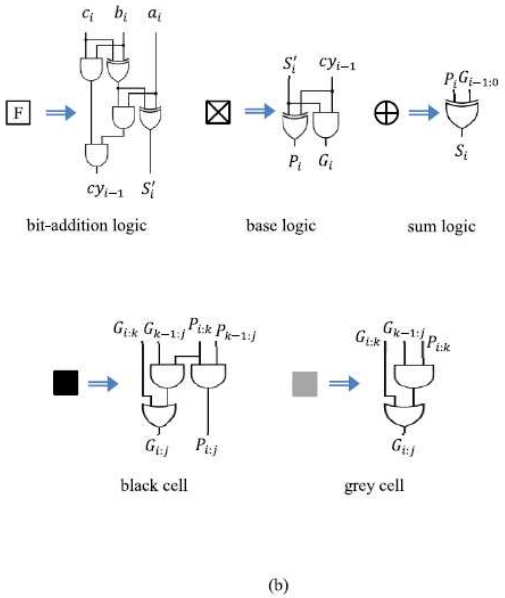


Fig 1. Proposed three-operand adder; (a) First order VLSI architecture cell.



(b)

Fig 1. (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 consideration for three-operand addition in the proposed adder technique. This additional carry-input signal (Cin) is taken as input to base logic while computing the G0 (S0 r Cin) in the first saltire-cell of the base logic. The third stage is the carry computation stage called "generate and propagate logic" (PG) to precompute the carry bit and the combination of black and grey cell logics. The logical diagram of black and grey cell is shown in Fig. 2 that computes the carry generate Gi j and propagate Pi j signals.

3.2. Modified Dual-CLCG Method:

The proposed modified dual-CLCG method generates pseudorandom bits by congruential modulo-2 addition of two coupled linear congruential generator (CLCG) outputs and is mathematically defined as follows,

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 obtained by using the congruential modulo-2 equation 1,

$$Z_i \equiv (B_i + C_i) \bmod 2 = B_i \oplus C_i$$

Where

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

Here, a1, b1, a2, b2, a3, b3, a4 and b4 are the constant parameters; x0, y0, p0 and q0 are the initial seeds. The necessary conditions to get the maximum length period are same as the existing dual-CLCG method (as discussed in Section-II). The proposed modified dual-CLCG method uses the congruential modulo-2 addition of two different coupled LCG outputs as specified in equation (1). Hence, the congruential modulo-2 addition does not skip any random bits at the output stage and produces one-bit random output in each iteration. Since, the coupled LCG has the maximal period, the modulo-2 addition of two coupled-LCG outputs in the modified dual CLCG have also the same

maximum length period of 2n for n-bit modulus operand. To perform the modulo-2 addition operation, it takes only single XOR logic. The proposed PRBG method can reduce the large memory area used in the existing dual-CLCG method and also can achieve the full-length period of 2n.

4. Results

Result of the proposed design is implemented using Xilinx ISE for simulation and Synthesis.

16 BIT 3-OPERAND ADDER:
Simulation:

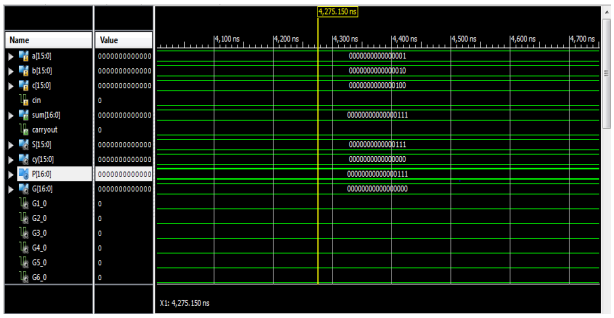


Fig. 2 Simulation.

Synthesis Result:

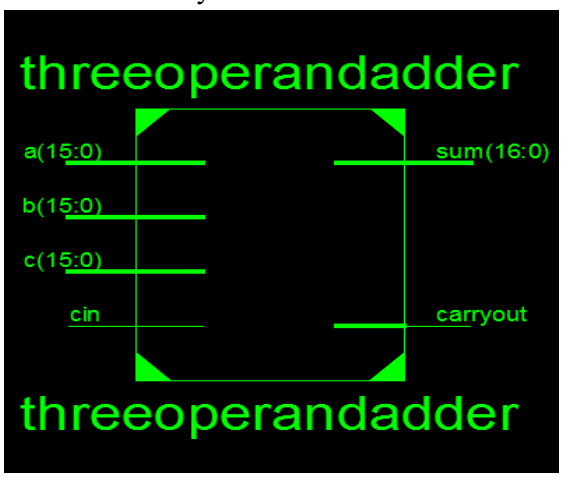


Fig 3. RTL Schematic.

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 4. 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 5. Timing Summary.

Proposed Results

32-BIT 3-OPERAND ADDER:
Simulation:

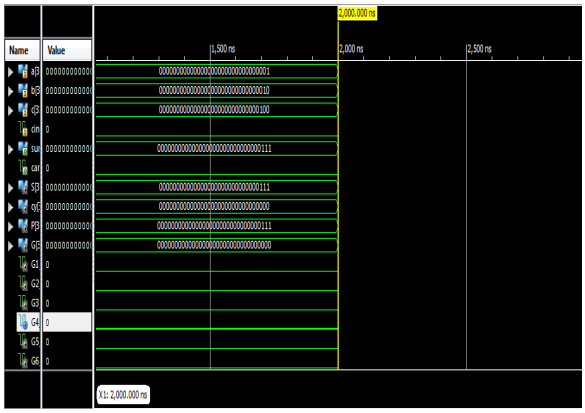


Fig 6. Simulation.

Synthesis Result:

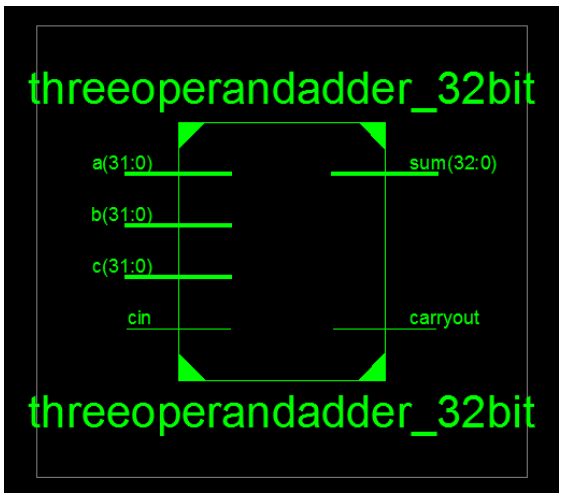


Fig 7. RTL Schematic

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 8. 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 9. Timing Summary.

MODIFIED DUAL CLCG RESULT:

Simulation:

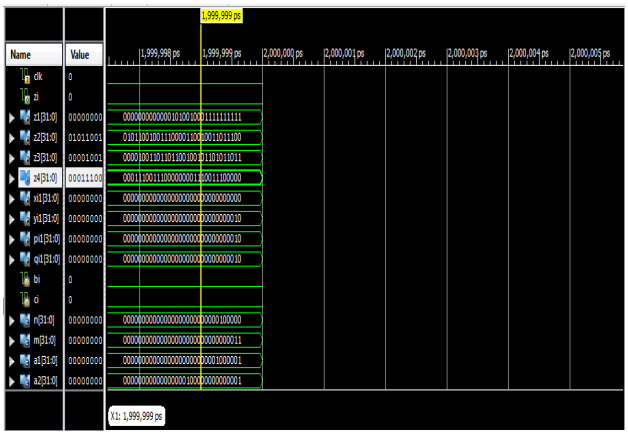


Fig 10. Simulation**.**

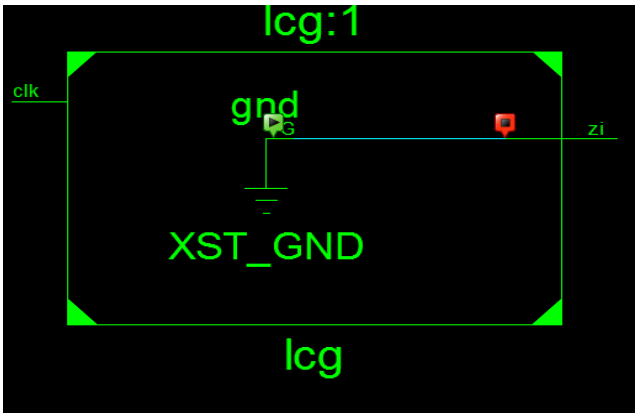


Fig 11. RTL Schematic

5. Conclusion

In this paper, a three-operand binary addition technique and its VLSI architecture are proposed for efficient computation of modular arithmetic used in cryptography and PRBG applications. This proposed design is unique in that it reduces delay and area in the prefix computation stages of PG logic and bit-addition logic, resulting in a reduction in critical path delay, area-delay product (ADP), and power-delay product (PDP). Furthermore, the proposed adder architecture is used to replace the CS3A three-operand adder architecture in a 32-bit MDCLCG architecture (previously published in the literature), and the design is prototyped on a commercially available FPGA platform to validate the design on a silicon chip.

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] A. Kumar Panda and K. Chandra Ray, "A coupled variable input LCG method and its VLSI architecture for pseudorandom bit generation," IEEE Trans. Instrum. Meas., vol. 69, no. 4, pp. 1011–1019, Apr. 2020.

[3] A. 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.

[4] K. S. Pandey, D. K. B. N. Goel, and H. Shrimali, "An ultra-fast parallel prefix adder," in Proc. IEEE 26th Symp. Comput. Arithmetic (ARITH), Kyoto, Japan, Jun. 2019, pp. 125–134.

[5] F. Jafarzadehpour, A. S. Molahosseini, A. A. Emrani Zarandi, and L. Sousa, "New energy-efficient hybrid wide-operand adder architecture," IET Circuits, Devices Syst., vol. 13, no. 8, pp. 1221–1231, Nov. 2019.

[6] 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.

[7] 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.

[8] 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 Integer. (VLSI) Syst., vol. 24, no. 2, pp. 434–443, Feb. 2016.

[9] S. S. Erdem, T. Yanik, and A. Celebi, "A general digit-serial architecture for Montgomery modular multiplication," IEEE Trans. Very Large-Scale Integer. (VLSI) Syst., vol. 25, no. 5, pp. 1658–1668, May 2017.

[10] A. Rezai and P. Keshavarzi, "High-throughput modular multiplication and exponentiation algorithms using multibit-scan–multibit-shift technique," IEEE Trans. Very Large-Scale Integer. (VLSI) Syst., vol. 23, no. 9, pp. 1710–1719, Sep. 2015.