ISSN (Online): 2583-7052

International Journal of Advanced Trends in Engineering and Management (IJATEM)
Vol. 03 (05), June2024, pp. 58-68

Article Title: Implementation of a Reverse Carry Propagate Adder Based On Efficient VLSI for Hardware Security and Area Consumption

Implementation of a Reverse Carry Propagate Adder Based On Efficient VLSI for Hardware Security and Area Consumption

S. Nagakumararaj

PhD Scholar, Department of Electronics and Communication Engineering, Karpagam Academy of Higher Education, Coimbatore. kumarrs2005@gmail.com

ABSTRACT

The various techniques inbuilt in the VLSI (Very Large Scale Integration) designing is originated from the basis of adder circuits. When these adder circuits are processed in the higher digital signal processing systems, the adders have to exhibit high performance, good efficiency with reduced cost and area of implementation. The major objective of designing the VLSI design is minimizing the complexity and to perform the operations more effectively. Whereas in some other methods, the logic circuit designing is made larger in order to achieve higher efficiency. This in turn, results in the reduction of hardware security and makes the computation more complex. This paper utilizes the RCPA adder logic to reduce the area for the implementation and enhancing the hardware security. By implementing RCPA adder circuit, the carry propagates in the opposite direction from the most significant bit to the least significant bit. The usage of shift accumulator improves the speed of process and reduces the area required. By implementing the reverse propagation of the carry in the adder circuit, computation time is optimized and the implementation is made efficient with the reduction in the delay.

Keywords: VLSI (Very Large Scale Integration), Reverse carry propagation adder (RCPA), hardware security, area consumption.

1 Introduction

Very large scale integration are the combination of enormous number of logic gates especially the adder circuits. These binary adder circuit are the basic logic gates involved in implement various arithmetic logics. It is be notable that the performances of these adders is more reliable for the effective designing and operations of the digital circuits [1]. In most of these arithmetic logic implementation, Full Adder (FA) is used in designing the circuits. The FA is used in performing addition, subtraction and multiplier operations in more effective manner. Even for large scale logic implementation, efficient full adder block is deployed [2].

In any basic adder logic, the three binary operand addition is initially performed. For the three operand addition, carry save adder is implemented [3] which is also used for higher number of bits. Adders are also used in Digital Signal Processing (DSP) for image processing and video processing. For the effective signal processing, adders thus consume less power with less

propagation delay. The propagation delay is based on the number of logic gates implemented and also the number of input and output loads involved [4]. By reducing the propagation delay, the efficiency is increased.

One of the most specified adder is the ripple carry adder, where less power consumption is involved. Ripple carry adder is mostly the combination of four full adders where the carry get rippled from one adder to the other. These ripple carry adders are used in design very complex logics and enhances the optimization for those complex logic circuits [5]. In the ripple carry adder several adder are connected next to each other for the number of logic gates used. But usage of ripple carry adder increases the power dissipation leakage, thus reduces the circuit performances [6].

There exists delay in the ripple carry adder as the resent output depends on the previous output, which is overcome by using the carry look ahead adder. This existing delay is overcame by installing fast performing logic gates like carry look ahead adders [7]. In carry look ahead adders logic functions like propagating and generating are included, by with the carry propagating delay is reduced to exhibit best performance with less time delay. This improves the speed of processing in these adders [8]. Further improvement is attained by designing the approximate carry look ahead adder (ACLA) by reducing the error probability in the carry generating logic thereby reducing the number of logic gates included. The cascaded carry look ahead adder requires less energy to produce image with better quality, thus this adders is implemented in the digital image processing techniques [9]. These adders is implemented in data processing like data encryption, decoding, decryption as it comprises of ALU and MAC units [10]. But the carry look ahead adder includes lots of computations, and hardware complexity.

To improve the efficiency of logical operations by maintaining low delay in less occupied area, RCPA is implemented. By propagating the carry bit in reverse direction i.e., from the most significant bit (MSB) to the least significant bit (LSB), the delay is reduced. On the other hand, RCPA also requires only less area and power for its implementation. This also minimizes the unwanted partial product thus makes the computations simpler. Hence, the suggested method reduces the delay, area and makes the computation easier, and thus enhances the efficiency and the accuracy of this method.

2 Recent works

Tari *et al* [2019] [11] have proposed full adder implementation based on the Carbon Nanotube Field Effect Transistor (CNT-FET) technique. This process is deployed for optimizing the power–delay product and delay in various load and supply. The performance and power consumed is enhanced as the full adder is implemented in nano-scale. In this process two paths procedures are involved, initially the inputs are generated and then the source and the inputs are connected to the P-CNTFET and the N-CNTFET. This circuit increases the speed of

processing and makes the construction simpler, en in designing multi-bit adders but full adder designed using carbon nanotube results in lot of impurities thus reduced the efficiency.

Lee *et al* [2021] [12] have developed cell-based full adder based on the Error Reduced Carry Prediction Approximate Adder algorithm. This adder implementation predict the carry and demolishes the error during with the error reduction process. This error reduction process is carried out by reducing the error distance (ED) where the carry propagation is not processed in the same order that is not from the lower order to the higher order bits. Along with the accuracy, the hardware efficiency is also improved. Finally, the effect of the error is reduced in the end output and the accuracy of this process is improved but even when the delay is reduced, the size is large comparatively.

Thakur *et al* [2020] [13] have analyzed different form of adders like Kogge stone (KS) adders, Carry-look ahead (CL), Carry-skip (CSK), Ripple-carry (RC) thus in order to enhance the computation and the speed of the processing. These adders are highly deployed in the VLSI designing in recording the speed and space required during implementation. Various algorithms for enhancing the speed of adders are already proposed but enhancing the speed in computation is processed for applications with large bit processing. For higher bit processing two addition techniques are used, In serial addition, block waits until it receive carry from the last one, but in parallel addition, the blocks do not waits for receiving the ripples. Even though reduction in area is implemented, involving serial adders cause delay in processing.

Jothin *et al* [2021] [14] have demonstrated square-root carry select adder (SQRT CSLA) in order to increase the speed and optimize the space and the efficiency. For this implementation, two methods are designed. As the first design, involving the design of carry enable binary to excess-1 converter (CEBEC) design, SQRT CSLA based on the binary to excess-1 convertor with enabling carry is implemented. By this, the operational speed is improved by optimizing the path of carry propagation. After the first adder designing, a regular SQRT CSLA is designed for reducing the number of gates used to make the implementation simpler. Even when the operational speed is improved, as CSLA reflects the operation of RCA, it undergoes some delay during the carry propagation which reduces the system efficiency.

Balasubramanian *et al* [2022] [15] have introduced architecture of carry look ahead adder to enhance the speed and makes the process more energy efficient. To tackle the worst case delay, the linear time in the carry look ahead adder is reduced as logarithmic time addition with the information of the carry input. The carry is made to propagate from the lowest position to the N-bit adder position, thus speeding up the processing time internally. As the power and the delay are optimized, the product of power and delay (PDP) is also low by which the efficiency is increased. This architecture decrease the delay and complexity due to the critical path delay but as this implementation involves lots of hardware components, the system implementation is costly.

ISSN (Online): 2583-7052

International Journal of Advanced Trends in Engineering and Management (IJATEM)
Vol. 03 (05), June2024, pp. 58-68

Article Title: **Implementation of a Reverse Carry Propagate Adder Based On Efficient VLSI for Hardware Security and Area Consumption**

3 Proposed work

The RCPA is introduced in attaining high efficiency of the system by optimizing the delay, speed and space occupied. Figure 1 indicates the process flow diagram of the proposed methodology.

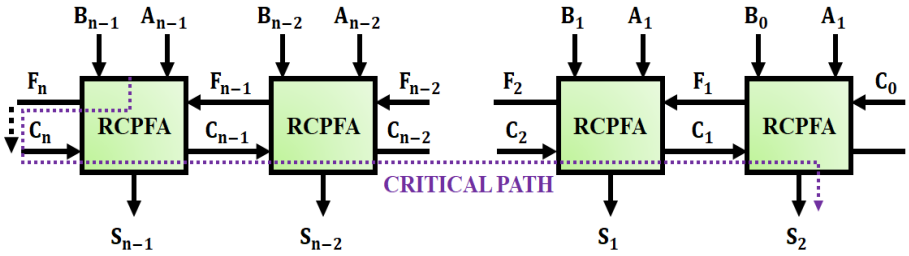

Figure 1: *Carry Propagation Adder*

The RCPA operates by propagating the carry output in the reversed direction that is, from the MSB to the LSB. The operation of this RCP adder involves four inputs and three outputs respectively. The inputs of this adder are input operands A_i and B_i, the carry of the next bit ie, C_{i+1} and forecast signal (F_i). The outputs are the Sum S_i, Carry C_i and the next forecast signal (F_{i+1})

3.1 Proposed Reverse Carry Propagate Full-Adder Cell

The RCPA is made by the combination of several full adders. Each of these full adders produce the sum signal along with the carry signal is represented as,

$$2C_i + S_i = A_i + B_i + C_i \quad (1)$$

Here, A_i (B_i) represents the i^{th} number of bit used as the input of A (B) in their iteration. Thus, equation (i) is used in determining the output of the i^{th} bit for the i^{th} iteration using the given input A and B, to produce the output C_i

By simplifying equation (i) by solving the Ci values we get,

$$S_i + C_i = A_i + B_i - 2C_{i+1} \quad (2)$$

By noticing equation (2), it is observed that the current (i^{th}) output of the RCPA depend on the next ($i+1^{th}$) output of the carry and the inputs fed to the adder. This clearly shows that the RCPA depends on the next carry output, and the carry outputs are reversed and the adder operation is carried out.

ISSN (Online): 2583-7052

International Journal of Advanced Trends in Engineering and Management (IJATEM)
Vol. 03 (05), June2024, pp. 58-68

3.2 K-MAP Analysis

The K-Map is drawn for the RCPA circuit as shown below.

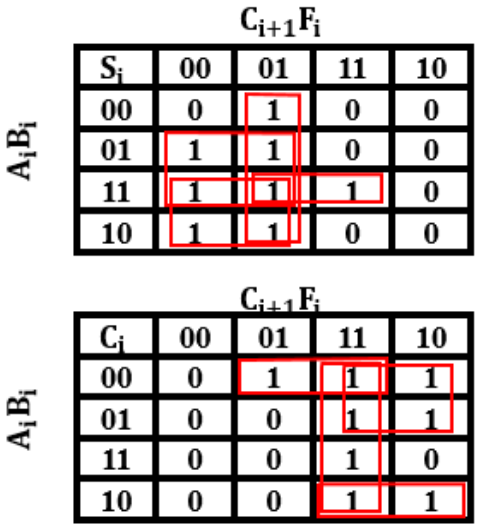

Figure 2: *K-MAP of outputs S_i and C_i for RCPA*

Figure 2 represents the K-Map for the sum and the carry bits for the RCPA circuit. This karnaugh map is used in determining the expression of the sum and carry of the RCPA including the forecast signal as the input fed, which makes the operation of the adder more efficient and produces accurate results.

$$S_i = \overline{C_{\iota+1}}F_i + \overline{C_{\iota+1}}A_i + \overline{C_{\iota+1}}B_i + A_iB_iF_i \quad (3)$$

$$C_i = C_{i+1}F_i + C_{i+1}\overline{A}_\iota + C_{i+1}\overline{B}_\iota + \overline{A_\iota B_\iota}F_i \quad (4)$$

Equation (3) represents the Boolean relation of the Sum for the i^{th} iteration with those input fed using the RCPA adder. Equation (4) represents the Boolean relation of the carry for the i^{th} iteration with those input fed using the RCPA adder.

By solving equation (3), we obtain equation (5) as,

$$S_i = F_i(\overline{C_{\iota+1}} + A_iB_i) + \overline{C_{\iota+1}}(A_i + B_i) \quad (5)$$

By solving equation (4), we obtain equation (6) as,

$$C_i = F_i\,\overline{(\overline{C_{\iota+1}}(A_\iota + B_\iota))} + \overline{(\overline{C_{\iota+1}} + A_\iota B_\iota)} \quad (6)$$

By generalizing the equations (5) and (6) by replacing the values as, $X_i = \overline{(\overline{C_{\iota+1}} + A_\iota B_\iota)}$ and$Y_i = \overline{(C_{\iota+1}}(A_i + B_i))$, we obtained the sum output Si as,

$$S_i = F_i\overline{X}_\iota + \overline{Y}_\iota \quad (7)$$

ISSN (Online): 2583-7052

International Journal of Advanced Trends in Engineering and Management (IJATEM)
Vol. 03 (05), June2024, pp. 58-68

And the carry output C_i as,

$$C_i = F_i Y_i + X_i \quad (8)$$

Thus equations (5) and (6) are the major equation for evaluating the RCPA adder operation along with the forecast signal and Equations (7) and (8) are the generalized form of these equations.

3.3 Error Analysis

As the carry propagation is done in the reversed order there occurs some error which is analyzed finely. These error analysis is carried out to detect the error and to rectify this to enhance the efficiency and the accuracy of the adders. In the RCP adder, the variation in the error follows the decrement in the bit significance. This implies that when the significance value of the bit increases, the collective effect of the error decreases. The major error factors occurs in the RCPA adder are Mean Relative Error Distance (MRED), Mean Error (μ), Mean Error Distance (MED) and the Variance (σ2).

3.3.1 Mean Relative Error Distance (MRED)

Mean Relative Error Distance (MRED) is the measure to detect the prediction error compared with the value of the normalization. The value of the MRED is always positive. This is mainly calculated for determining the accuracy of the adder circuit implemented.

$$MRED = \frac{1}{2^{2n}} \sum_{i=1}^{2^{2n}} \left|\frac{ED_i}{S_i}\right| \text{----------} \quad (9)$$

Equation (9) represents the formula to calculate the Mean Relative Error Distance (MRED) in the RCPA adder circuit where S_i is the sum result obtained and the ED_i is the error distance of the i^{th} input.

3.3.2 Mean Error (μ)

Mean Error (μ) is the comparison between the error occurred in the adder circuit performance and the original correct output. This is used to determine the efficiency of the adder performances.

$$\mu = \sum_{i=0}^{n-1} [P(e_i = 1) - P(e_i = -1)] \times 2^i \text{---------} \quad (10)$$

Equation (10) represents the mean value of the RCPA adder circuit where P () is the probability of the corresponding signal, e_i is the error in the i^{th} bit.

3.3.3 Mean Error Distance (MED)

Mean Error Distance (MED) is the mean value calculated from all the mean values of the Error Distance obtained from the output during the adder performance. Theses error distance occurs

when the main inputs of the adder circuit is not much determined and the produces certain error probability.

$$MED = \sum_{i=0}^{n-1} |ED_i| \times P(ED_i) \text{----------} \quad (11)$$

Equation (11) represents the Mean Error Distance (MED) occurred in the RCPA adder circuit performance where Ed_i is the error distance for the i^{th} bit, P () is the probability of the corresponding signal .

3.3.4 Variance (σ2)

Variance is used in simplifying the error parameter extracted in every full adders. This variance is used in determining the accuracy of the adder circuit performance. The variance of the circuit is computed by using the following equation.

$$\sigma^2 \cong \sum_{i=0}^{n-1} {ED_i}^2 \times P(ED_i) - \mu^2 \text{----------} \quad (12)$$

Equation (12) represents the Variance (σ2) occurred in the RCPA adder circuit performance where EDI is the error distance for the i th bit, P () is the probability of the corresponding signal, μ is the mean error calculated.

4 Results and discussion

The RCPA is implemented in the MATLAB simulation software. The output are recorded by varying the values of the input A, B and C_{in} fed to the adder circuit simulated. The result is obtained by implementing the RCPA adder in the Xilinx VLSI software.

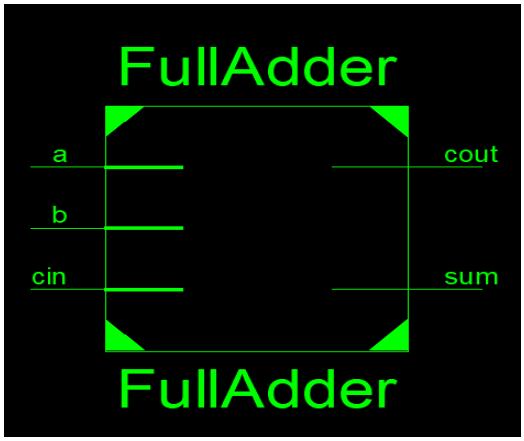

Figure 3: *Full Adder*

Figure 3 represents the full adder which is enormous times implemented together to build RCPA logic. A full adder comprises of three inputs along with the carry supplied and two output as the sum and the carry respectively.

ISSN (Online): 2583-7052

International Journal of Advanced Trends in Engineering and Management (IJATEM)
Vol. 03 (05), June2024, pp. 58-68

Article Title: **Implementation of a Reverse Carry Propagate Adder Based On Efficient VLSI for Hardware Security and Area Consumption**

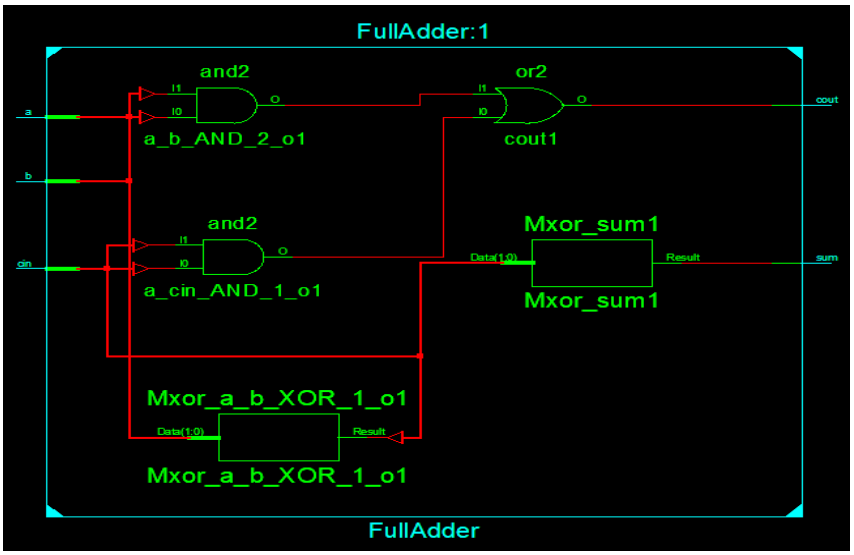

Figure 4: *Symmetric Diagram*

Figure 4 depicts the symmetric block diagram of the RCPA adder in the Xilinx VLSI software, which is simulated by including the basic logic gates in order to record the output changes in the Sum and Cout values.

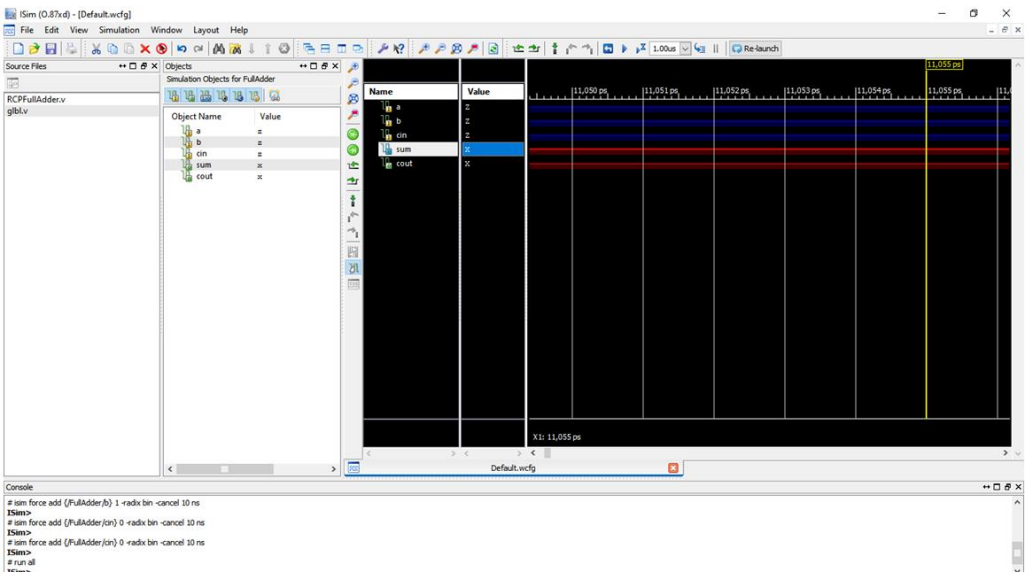

Figure 5: *Simulation Result (Initial Stage)*

Figure 5 represents the initial simulation result where the values assigned to the inputs are low and blue indicates the input and red line indicates the output. The result of this simulation produces low for the two outputs sum and cout.

ISSN (Online): 2583-7052

International Journal of Advanced Trends in Engineering and Management (IJATEM)
Vol. 03 (05), June2024, pp. 58-68

Article Title: **Implementation of a Reverse Carry Propagate Adder Based On Efficient VLSI for Hardware Security and Area Consumption**

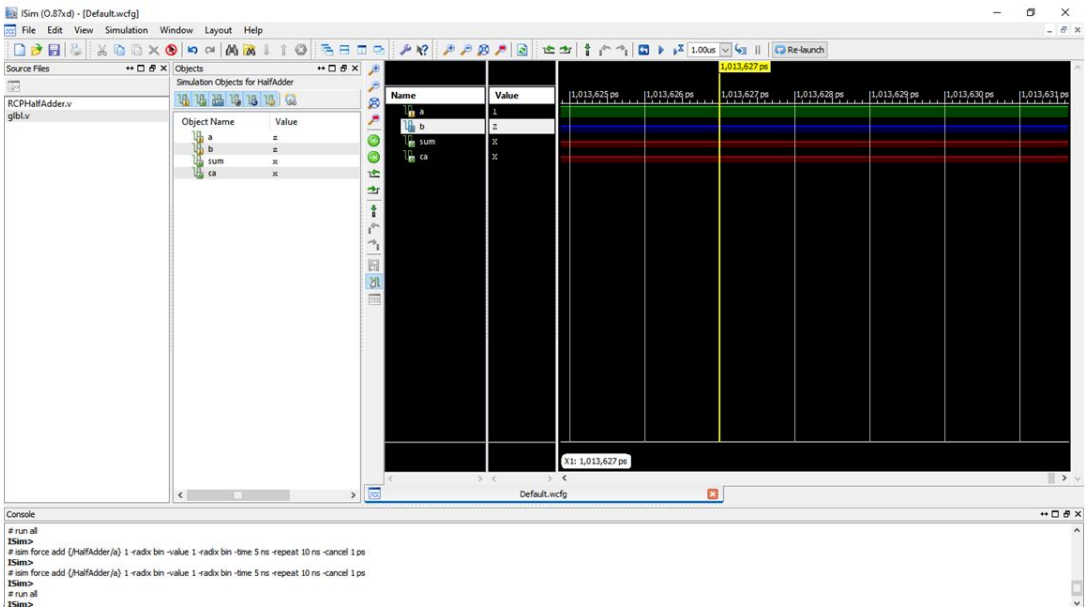

Figure 6: *Simulation Result (Progress Stage)*

Figure 6 represents the progressing stage where values are assigned to the input values to analyze the performance of RCPA adder, where the output for these input values depends on equations framed that depict the logic of RCPA adder circuits.

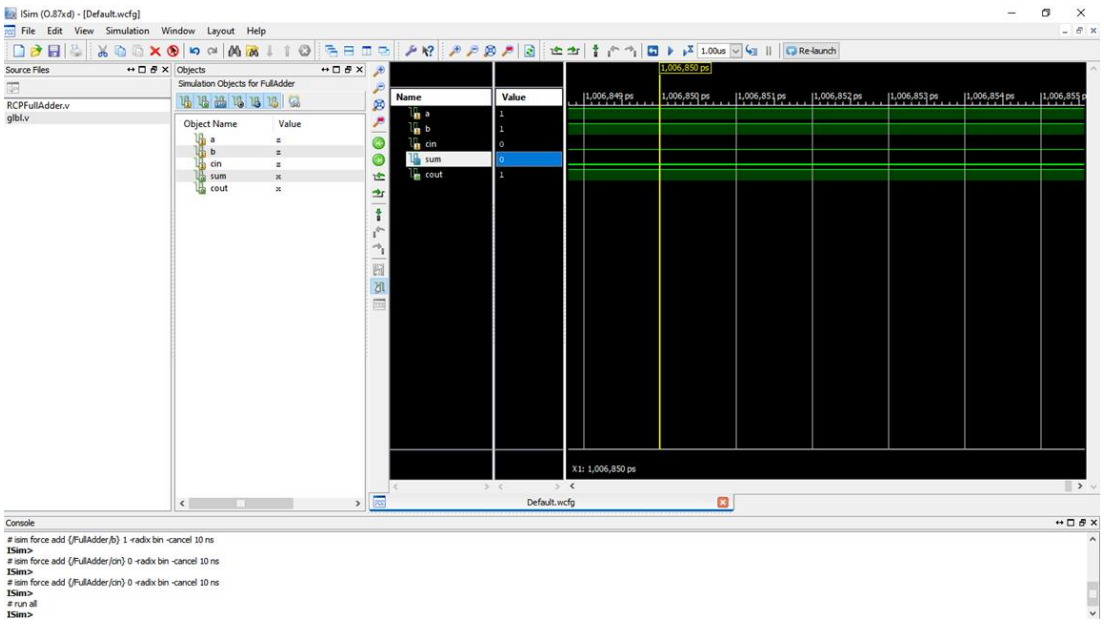

Figure 7: *Simulation Result (Final Stage)*

Figure 7 represents the operation of the RCPA adder. When the inputs A and B are assigned with high value input and C_{in} is assigned with low input, the output of sum is low and that of the carry is high.

5 Conclusion

By implementing VLSI design using RCPA, the area required for the designing is reduced and the hardware security is increased. As the RCPA follows the FIR filtering algorithm, the performance is more stable and efficient. This optimized implementation reduces the power consumption, as the process is reversed the delay in processing is reduced to great extent. As compared to other adder implementation using VLSI design, the implementation of RCPA reduced area consumption and it enhances hardware security.

References

1. Huang Junqi; Nandha Kumar; Haider Abbas, Year: 2020, "General Expressions for Performance Evaluation of Binary Adders", In 2020 17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), pp. 357-363. IEEE.

2. Singh, Salam Surjit; Dolly Leishangthem; Nasiruddin Shah; Biraj Shougaijam, Year: 2020, "A Unique design of hybrid full adder for the application of low power VLSI circuits", In 2020 4th international conference on electronics, communication and aerospace technology (ICECA), pp. 260-264, IEEE.

3. Panda Amit Kumar; Rakesh Palisetty; Kailash Chandra Ray, Year: 2020, "High-speed area-efficient VLSI architecture of three-operand binary adder", IEEE Transactions on Circuits and Systems I: Regular Papers, Vol: 67, no: 11, 3944-3953.

4. Joy; Mary Christina; Ansa Jimmy; Tony Thomas; Manju Kollannur, Year: 2020, "Modified 16 bit carry select and carry bypass adder architectures for high speed operations", In 2020 IEEE International Conference for Innovation in Technology (INOCON), pp. 1-4. IEEE.

5. Vahabi Mohsen; Ali Newaz Bahar; Akira Otsuki; Khan Wahid, Year: 2022, "Ultra-low-cost design of ripple carry adder to design nanoelectronics in QCA nanotechnology", Electronics, Vol: 11, no: 15, pp. 2320.

6. Padmini; Rajesh; Raghu; Megha Sree; Apurva, Year: 2021, "Design and analysis of 8-bit ripple Carry Adder using nine Transistor Full Adder", In 2021 7th international conference on advanced computing and communication systems (ICACCS), Vol: 1, pp. 1982-1987. IEEE.

7. Hasan Mehedi; Parag Biswas; Shihabul Alam; Hasan Zaman; Mainul Hossain; Sharnali Islam, Year: 2019, "High speed and ultra-low power design of carry-out bit of 4-bit carry look-ahead adder", In 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp. 1-5. IEEE.

8. Muralidharan; Sathish Kumar, Year: 2020, "Design and implementation of low power and high speed multiplier using quaternary carry look-ahead adder", Microprocessors and Microsystems, Vol: 75, pp. 103054.

9. Garg Bharat; Sujit Kumar Patel, Year: 2021, "Reconfigurable carry look-ahead adder trading accuracy for energy efficiency." Journal of Signal Processing Systems, Vol: 93, pp. 99-111.

10. El-Bendary; Mohsen; Ayman Haggag, Year: 2021, "Investigating performance analysis of a novel low-power efficient area carry-look ahead adder", In 2021 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC), pp. 364-369, IEEE.

11. Tari Hani Taheri; Arezoo Dabaghi Zarandi; Mohammad Reza Reshadinezhad, Year: 2019, "Design of a high performance CNTFET-based full adder cell applicable in: Carry ripple, carry select and carry skip adders", Microelectronic Engineering, Vol: 215, pp. 110980.

12. Lee Jungwon; Hyoju Seo; Hyelin Seok; Yongtae Kim, Year: 2021, "A novel approximate adder design using error reduced carry prediction and constant truncation", IEEE Access, Vol: 9, pp. 119939-119953.

13. Thakur Garima; Harsh Sohal; Shruti Jain, Year: 2020, "Design and analysis of high-speed parallel prefix adder for digital circuit design applications", In 2020 International Conference on Computational Performance Evaluation (COMPE), pp. 095-100. IEEE.

14. Jothin; Sreelatha; Ahilan; Peer Mohamed, Year: 2021, "High-performance carry select adders", Circuits, Systems, and Signal Processing, Vol: 40, no: 8, pp. 4169-4185.

15. Balasubramanian Padmanabhan; Nikos Mastorakis, Year: 2022, "High-speed and energy-efficient carry look-ahead adder", Journal of Low Power Electronics and Applications, Vol: 12, no: 3, pp. 46.