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

VLSI Implementation of High-Performance Ternary Operand PPA for Fir Filter Application

Vaishnavi Kapilavai[1]*, Radha Krishna Vadde[2]

[1]M.Tech, Digital Electronics and Communication Engineering, G. Narayanamma Institute of Technology and Science, India

[2]Assistant Professor, Department of ECE, G. Narayanamma Institute of Technology and Science, India

Article Info
Volume 9, Issue 4
Page Number : 657-663

Publication Issue
July-August 2022

Article History
Accepted : 05 August 2022
Published : 29 August 2022

ABSTRACT

The binary digits adder, which performs mathematical operations, is the fundamental operational element of several cryptographies as well as pseudo-random bit generator approaches. The ripple carry adder, the final CSA step, requires an additional delay to operate. The parallel prefix adders use more space even though performance in terms of latency is improved. Parallel prefix adders can also be used to build three operand adders. A novel, low delay, area-efficient adder has been developed to improve efficiency in terms of delay as well as area strategy is utilized. FIR Filter having the element of adders and multipliers as a design parameter. So, the main objective of the DSP field is to reduce the area and delay in filter designs. This design is less complex in terms of area hardware and delay than similar designs that have been used in the past. Using the Xilinx ISE 14.7 version tool, the findings of the performance study and simulations are validated.

Keywords: Carry save adder, Three operand adder, Pseudo Random number Generator, cryptography.

I. INTRODUCTION

In modulo, which would be widely applied in cryptography, three operand adding are the most typical. Cryptography techniques must be deployed on hardware to ensure both physical security and optimal system efficiency [1]. Modular arithmetic, with the three-operand adder as the fundamental building block [3], is employed in cryptography applications [2] [4]. Three operand addition is necessary in these applications that use modular arithmetic; it is a basic operation in just this field. It is therefore imperative to create an effective three operand adder.

Data privacy in internet services could become a pressing issue to be resolved given the rapid evolution of data communication. Cryptographic software is mostly used to offer privacy. Modulo as well as cryptographic applications both use three operand adders as their essential building block. As a result, the fundamental components used determine the overall efficiency of Module Arithmetic as well as Cryptographic systems. The top module's functionality is dependent on this foundational module.

A two-operand addition is one that performs addition among two operands, hence two n-bit input values. A variety of adders, including rippling carry adders (RCA), parallel prefixed adders (PPA), carry skipping adders (CSKA), and others, are offered for use in two-operand operations.

A ripple carry adder is an example. When adding two operands, this is the adder that is most frequently employed. The 1-bit full adder units are simply cascaded in order to construct it. Its critical path latency is the main flaw of this adder. Before performing its initial full adder operation, the 2nd adder must wait. Likewise, it must wait until after the (n-1) operation before executing n-bit addition. So, with a ripple carry adder, the delay is greater.

Carry-save adder (CS3A) occupies less region and frequently used algorithm for performing the 3-operand addition in modular arithmetic operations applicable for cryptographic protocols and PRBG approaches [7]. Due to the greater delay in transmission in ripple-carry phase, has a significant impact on MDCLCG. Minimize critical path latency for expanding area with three-operand binary addition by using the parallel prefixed 2-operand adder known as Han-Carlson (HAN-CARLSON). According to this work, a new pre-compute bit - wise 3-operand adder that really is quick as well as space-efficient accompanied by carry-prefix calculation analogy to perform the 3-operand addition as well as consumes much less gate area as well as transmission delay than the HAN-CARLSON relied 3-operand adder.

PRBG is a fundamental unit of getting data during transmission from one place to other, which can be used in different cryptography applications [5]. However, this method encounters a few drawbacks like uses more memory utilization and high latency in initial clock, greatly decreasing in randomness. The implemented adders of area and power delays are tested especially in comparison with the 3-operand CSA and HAN-CARLSON adder approaches. The objective of this research is to design a 3-operand adder with less critical path delay and less complexity than other previous designs.

The remaining portion of this project work is ordered as follow: The carry-save as well as Han-Carlson are discussed in Section II and various adder architectures that are designed previously. Section-III emphasizes the proposed implementation of modified three-operand binary adder as well as the Random pattern generation applications. The section IV includes the suggested adder synthesis results as well as other existing adder approaches and comparison between them. The suggested adder is also included in the updated Modified Random pattern application to test the quantifiable metrics and to validate the findings using post-synthesis simulated outcomes. Additionally, the suggested design Synthesized on Various FPGA boards for various performance analyses. Section V brings the paper to a conclusion.

II. PREVIOUS WORK

The 3-operand binary addition is a key arithmetic operation in extent to which an individual modular arithmetic schemes including LCG-based PRBG techniques such as CLCG [6], MDCLCG, as well as CVLCG [8]. One stage of 3-operand adders or two stages of 2-operand adders can be utilised to put it into action. People typically use the carry-save adder to achieve 3-operand binary adding (CSA) [9]. With three or more operands, it conducts a two-stage addition. The array of complete adders is the initial stage. The second step of addition, known as the ripple-carry adder, outputs the total and carry of the operation. As a result, as the bit length increases gradually, the delay increases owing to the carry propagation adder in the last stage, the result is linear.

The critical path latency can be decreased by utilising a parallel prefixed 2-operand adder for 3-operand binary addition, such as Han-Carlson. When compared to other parallel prefix adders, the Han-Carlson adder is among the fastest.

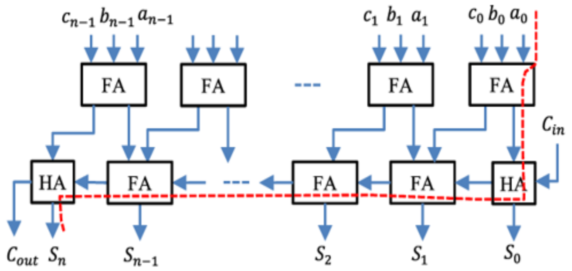


Figure 1. Carry Save Three Operand adder

Han-Carlson adders have straightforward prefixed leaf unit. An illustration of a divide as well as conquer plan is the Han-Carlson. This Arrangement adds two smaller adders at each level to calculate prefixed for 2-bit groupings, 4-bit groups, 8-bit groups, 16-bit groups, and so on. Generation step of the Han-Carlson adder: By computing the huge group prefixes as well as the interim prefix bits, the divide and conquer principle provides the latency of log2n stages. The advantage of this adder is architecture is simple and regular but fan-outs double at each stage it has fan out problem.

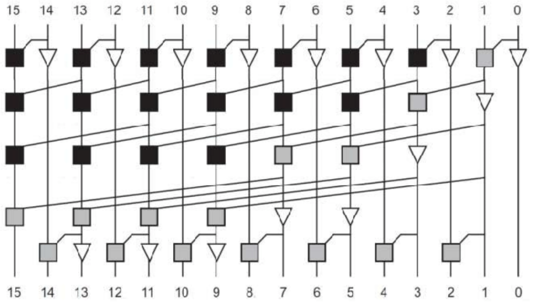


Figure 2: Architectural Diagram for Ladner Fischer adder

Because it decreases the latency to (n/2) +1 stages, the Han-Carlson tree is popularly known as the divide-and-conquer tree. Along with the large group prefixes, it computes intermediate prefixes. At each stage, the fan outs are doubled. Wide adders perform poorly as a result of the increased fan outs. If the fan-out gates are large enough or the appropriate signals are delayed before would be used for the intermediary prefixes, we can see a boost in speed. Because each cell must be several sizes, transistor sizing disrupts the design layout's regularity. The bigger gates, on the other hand, can be extended across neighboring columns. With the right buffering, the fan outs could be controlled.

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

Pre-processing stage is used to find the propagate and generate values and post processing stage is used to calculate the sum value which are shown in figure 2. The carry generation is different for different parallel prefix adder.

Prefixes for two-bit groups are computed using the Brent Kung adder. Prefixes of four-bit groups, for example, can be found using these prefixes. The carry out of the given bit step is then calculated using these prefixes. These carries will be combined with the following stage's Group Propagate to compute the stage's Sum bit. Each bit level has a maximum fan-out of two. The fan-out will be limited, and the loading on successive levels will be lowered. The carry chain is binary addition's major issue. The width of the input operands as well as the carrying chain's length increase simultaneously. It is feasible to accelerate, but not remove, the carry chain in order to increase the performance of the carry-propagate adders. As a consequence, because they tend to determine the critical route with most calculations, most digital designers end up constructing faster adders via improving computer architecture, as a result it will improve the area hardware complexity as well as critical path delay for 3-operand binary adder

III. PROPOSED METHOD

Finite impulse response (FIR) filters find wide applications in digital systems ranging from multimedia signal processing to wireless mobile communications. FIR filters are employed in high sample rate applications like video processing. On the other hand, applications like MIMO (multiple inputs multiple outputs) systems used in cellular wireless communication have the stringent requirement of a low power FIR filter circuit which operates at moderate sample rate.

FIR Filter having the elements of adders, multipliers along with delay elements. So, in order to get some better performance in terms of area and delay we can choose various types of multipliers available in existing.

Wallace Tree Multiplier

Multipliers have gained the significant importance with the introduction of the digital computers. Multipliers are most often used in digital signal processing applications and microprocessors designs. In contrast to process of addition and subtraction, multipliers consume more time and more hardware resources. Numerous multiplication approaches have been put into practise recently thanks to technological advancements in order to provide multipliers that are high speed, low power, small in size, or a combination of all three. The two primary restrictions that compete with one another are speed and area. Therefore, it is the designer's task to decide proper balance in selecting an appropriate multiplication technique as per requirements. Parallel multipliers are the high-speed multipliers. Therefore, the enhanced speed of the multiplication operation is achieved using various schemes and Wallace tree is one of them. There are three phases in the multiplier architecture:

1. The first phase is the generation of partial products;
2. Accumulation of partial product in second phase; and
3. The third phase is the final addition phase.

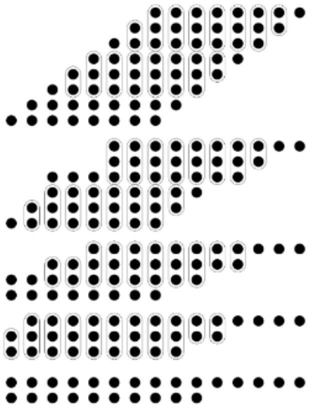


Figure 3: 8x8 Wallace tree architecture

Booth Multiplier:

Booth algorithm gives a procedure for **multiplying binary integers** in signed 2's complement representation **in efficient way**, i.e., a smaller number of additions/subtractions required. It operates on the fact that strings of 0's in the multiplier require no addition but just shifting and a string of 1's in the multiplier from bit weight 2^k to weight 2^m can be treated as 2^ (k+1) to 2^m. As in all multiplication schemes, booth algorithm requires examination **of the multiplier bits** and shifting of the partial product. Prior to the shifting, the multiplicand may be added to the partial product, subtracted from the partial product, or left unchanged according to following rules:

1. 1. Whenever a string of 1s in the multiplier contains a least significant 1, the multiplicand is deducted from the partial product.
2. 2. When the multiplicand comes across the first 0 in a string of 0s in the multiplier (assuming there was a prior "1"), it is included to the partial product.
3. Whenever the multiplier bit is the same as the preceding multiplier bit, the partial product remains unchanged.

Best Case and Worst-Case Occurrence:

Best case is when there is a large block of consecutive 1's and 0's in the multipliers, so that there is minimum number of logical operations taking place, as in addition and subtraction. Worst case is when there are pairs of alternate 0's and 1's, either 01 or 10 in the multipliers, so that maximum number of additions and subtractions are required

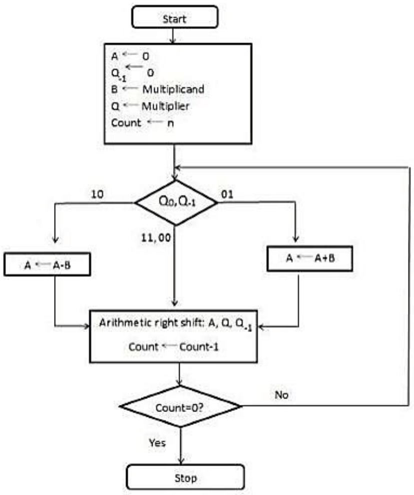


Figure 4: Flow Chart of Booth Multiplier

Wallace-Booth Algorithm:

Modified Radix4 Booth algorithm which scan strings of three bits with the algorithm given below:

1) Extend the sign bit 1 position if necessary to ensure that n is even,

2) Append a 0 to the right of the LSB of the multiplier,

3) According to the value of each vector (see ensure that n is even. each Partial Product will he 0, +y, -y, +2y or -2y.

The negative values of y are made by taking the 2's complement and in this paper cany-look-ahead (CLA) fast adders are used.

By moving y one bit to the left, the number y is multiplied. In any event, only n/2 partial products are produced when designing n-bit parallel multipliers.

Wallace used the following methodology, which may be summed up as follows: A series of Carry-Save-Adder (CSA) adders is used to minimise the partial products once they have been generated.

Full adders are gathered into a CSA. Three n-bit operands are accepted by the CSA, which produces two n-bit results—an n-bit carry and an n-bit partial summation. These two bit-sequences are fed into a second CSA, which creates a new partial sum carry after receiving them along with another input operand. The term carry-propagate adder (CPA) is used to denote an adder which is not CSA. A propagate adder may propagate its carry using ripple-carry adders, carry-look-ahead (CLA) or some other methods and in this design, CLA is used. In general case, the delay accumulation of ripple carry-adders.

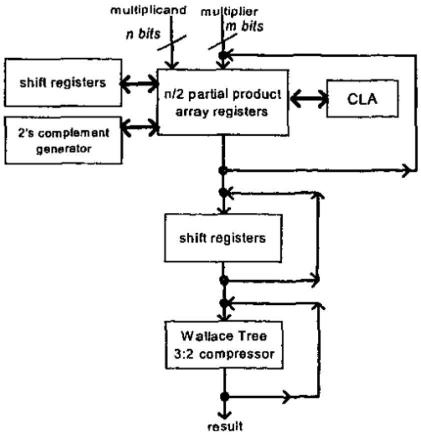


Figure 5: Flow Chart of Wallace Booth Multiplier

IV. Results

We have to give force constant values to our design once the waveform window gets open, otherwise the second option we have to provide some different input pairs as a test module and we can run the module directly to cross verify the results.

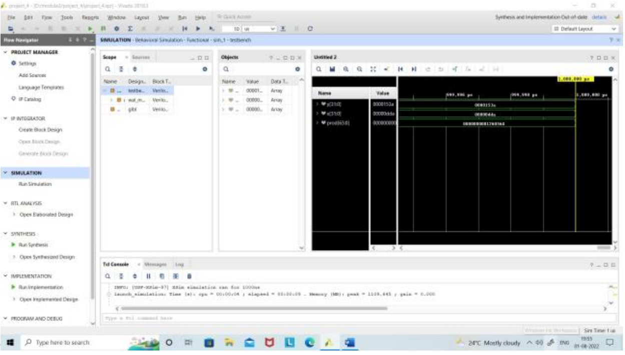
3 

Figure 6. Simulation outcomes for Wallace Multiplier

The simulation Results of Wallace tree multiplier is shown in fig 5.

Compared to Wallace tree multiplier, Booth multiplier has clock and reset as a control signal. So if we provide positive edge clock signal means it will do the operation when clock goes from 0 to 1, While the operation will take place when the clock swings from 1 to 0 if the edge clock signal is negative. The same instruction also followed for reset condition too.

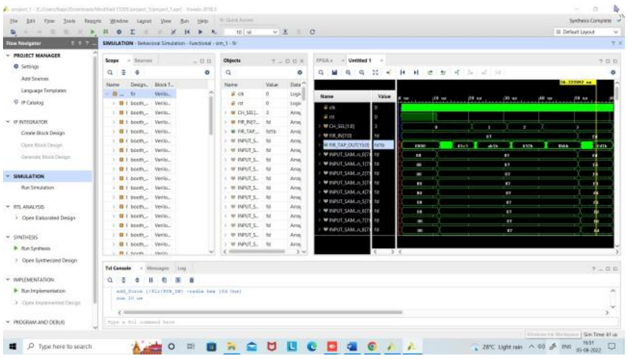


Figure 7. Simulation outcomes for Booth Multiplier

The simulation Results of Booth multiplier is shown in fig 6

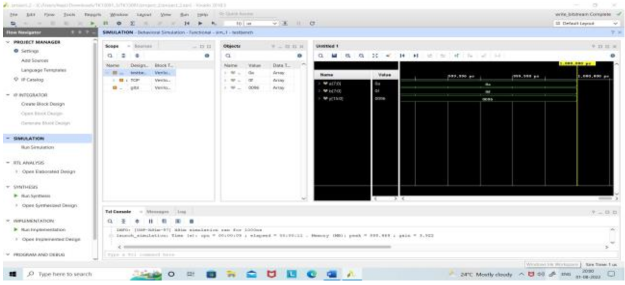


Figure 8. Simulation outcomes of Wallace Booth Algorithm

By Comparing the all results, Booth Wallace tree multiplier have the better performance in terms of area and delay as well.

Table.1. Comparison Table for various Multipliers

	Area (LUT's)	Power(W)	Delay
Booth	5453	0.135	34.39
Wallace tree	3290	0.097	104.7805
Booth Wallace	1506	0.086	76.5645

The above Table 1 shows the comparison between various Multipliers implemented in Nexys Artix-7 board in Xilinx FPGA for 32-bit width.

V. Conclusion

This work successfully implements an effective 3-operand binary adder. Three additional operands can be added using the tools. The three-operand binary adder is included in that has been proposed and it shows better outcomes in terms of area and delay. Verilog Hardware Description Language coding style is used to implement the CSA, HC3A, Booth Wallace tree multiplier in Nexys Artix-7 FPGA for 32 bits. The paper can be extended to implement an encryption and decryption algorithms like AES in future scope by using the Random number generator.

VI. REFERENCES

[1]. M. M. Islam, M. Shahjalal, and Y. M. Jang, FPGA implementation of excessive-speed vicinity-inexperienced processor for elliptic curve component multiplication over high subject, IEEE Access, vol. 7, pp. 178811–178826, 2019.

[2]. Z. Liu, and I. Verbauwhede, Elliptic curve cryptography with efficiently computable endomorphisms and its hardware implementations for the Internet of Things, IEEE Trans. Comput., vol. Sixty six, no. Five, pp. 773–785, May 2017.

[3]. S. S. Erdem, and A. Celebi, A popular digit-serial shape for montgomery modular multiplication, IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 25, no. Five, pp. 1658–1668, May 2017.

[4]. R. S. Katti and S. K. Srinivasan, Efficient hardware implementation of a brand new pseudo-random bit collection generator, in Proc. IEEE Int. Symp. Circuits Syst., Taipei, Taiwan, May 2009, pp. 1393–1396.

[5]. A. K. Panda and K. C. Ray, Design and FPGA prototype of 1024- bit Blum-Blum-Shub PRBG structure, in Proc. IEEE Int. Conf. Inf. Commun. Signal Process. (ICICSP), Singapore, Sep. 2018, pp. 38–forty three.

[6]. F. Jafarzadehpour, and L. Sousa, New strength-inexperienced hybrid massive-operand adder architecture, IET Circuits, Devices Syst., vol. Thirteen, no. Eight, pp. 1221–1231, Nov. 2019.

[7]. T. Kim, and S. Tjiang, Circuit optimization using carry-maintain adder (CSA) cells, IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 17, no. 10, pp. 974–984, Oct. 1998.

[8]. A. K. Panda and K. C. Ray, Modified dual-CLCG technique and its VLSI structure for pseudorandom bit era, IEEE Trans. Circuits Syst. I, Reg. Papers, vol. Sixty six, no. Three, pp. 989–1002, Mar. 2019.

[9]. A. Kumar Panda and K. Chandra Ray, A coupled variable input LCG method and its VLSI structure for pseudorandom bit generation, IEEE Trans. Instrum. Meas., vol. Sixty nine, no. Four, pp. 1011–1019, Apr. 2020.

Cite this article as :

Vaishnavi Kapilavai, Radha Krishna Vadde, "VLSI Implementation of High-Performance Ternary Operand PPA for Fir Filter Application", International Journal of Scientific Research in Science and Technology (IJSRST), Online ISSN : 2395-602X, Print ISSN : 2395-6011, Volume 9 Issue 4, pp. 657-663, July-August 2022.
Journal URL : https://ijsrst.com/IJSRST2294111