E-ISSN 2581 – 7957
P-ISSN 2277 – 3916

CVR Journal of Science and Technology, Volume 24, June 2023
DOI: 10.32377/cvrjst2404

Design and ASIC Implementation of Modified Shift-and-Add Algorithm using Redundant Arithmetic Integrator Adder and Subtractor

T. Subha Sri Lakshmi
Asst. Professor, CVR College of Engineering/ECE Department, Hyderabad, India
Email: rupashubha@gmail.com

Abstract: **Shift and Add Algorithm, also referred to as Volder's algorithm and the digit-by-digit technique. This is a specialized digital computer designed for airborne real-time processing. To calculate the trigonometric relationships involved in a plane coordinate rotation and conversion from rectangular to polar coordinates, a specific computational technique is used in this case. Additionally, the shift and add algorithm apply to modern systems, square roots, logarithms, and exponential expressions. Trigonometric functions are highly important in computation units; currently, many mathematical functions Sine, Cosine, Tangent, etc., by applying this approach, it is very simple to compute. Redundant arithmetic is used to lower the delay and boost the speed of operation. The adders play a significant part in the shift and add algorithm and carry propagation in adders causes the delay to increase quickly and slow down the speed of operation. Carry-propagation chains are produced by conservative operations like addition, multiplication, and subtraction. To rectify this issue, redundant number schemes are used. Using redundant numbers by speeding up mathematical operations. This technique is applied in signal processing and other areas. In this paper, an efficient and modified shift-and-add algorithm is designed, which is used to minimize the rotation angles. The main idea of the algorithm is to replace the carry select adder with an Integrated Redundant Arithmetic adder and subtractor, to achieve better latency and maximum throughput. By using redundant adders, the algorithm can be implemented with high speed and low power.**

Index Terms: **Shift and Add algorithm, Carry Select Adder (CSLA), Redundant Arithmetic Adder (RAA), rotation angles, Cadence-Innovus.**

I. Introduction

A common shift and add approach is used to compute a variety of arithmetic, logarithmic, trigonometric, and hyperbolic functions [1, 2]. The Shift and Add technique, which lowers complex multiplication and significantly reduces overall hardware complexity, is very straightforward and iterative. It is now a common option when a designer seeks to balance hardware requirements with latency. As a result, it was used in a wide range of industries including communications, multimedia, robotics, and the internet of things [3].

When compared to the traditional shift and add method, several variants have been suggested in the literature for an efficient shift and add implementation with the fewest repetitions. The reverse angle recoding is employed to get rid of the redundant shift and add elementary rotation [4]. The shift and add algorithm determine the number of iterations to do and compute the necessary angle using the fewest possible iterations. A redesigned shift and add architecture were presented that uses a new set of angles to split the rotational angles into smaller angles and calls for the least number of adders possible [5]. New, better designs were also suggested [6], replacing the regular adders in the architecture with area-efficient carry-select adders. With the enhanced power control and the hardware reduction approaches, a low-power and high-speed shift and add (LH-SAA) design is presented to produce a high-speed or the architecture of low latency VLSI for the shift and add algorithm [7].

In order to reduce the number of iterations and power consumption, hardware reduction techniques are proposed in this study along with an enhanced shift and add design that makes use of an integrated adder subtractor. The primary goal of the shift and add algorithm being suggested is to achieve low latency and high-speed VLSI design. The Hcub algorithm and the Canonical Signed-Digit (CSD) approach are used to reduce the number of shifters and adders/subtractors. These strategies are employed to carry out complicated processes. CSD implements the extension circuit's non-zero combination operations. Instead of being restricted by a constant bit, it creates a multiplier block out of the group of constants. The main contribution of the paper is an enhanced shift and add design that makes use of the built-in adder and subtractor.

II. Related Work

The revised shift and add algorithm, which replaces the shifts and adds micro-rotation by a new angle set, was introduced by Garrido et. al. [5]. Three new sorts of rotators are used in their novel strategy: friend angles, USR Shift & add, and nano-rotations. The proposed shift and add algorithm use the fewest adders among shift and add algorithms too far owing to the proposed micro-rotations.

Even though the new design has additional advantages, demonstrated in earlier research [6, 7] that it may still be improved architecturally by employing cutting-edge methods. Making the entire design power efficient with the architecture of the friend angles (5 Adder, 7 MUX, and 9) was the focus of the work's primary sections.

A shift and add with low power and high speed (LH-shift and add) is designed. A proposed design [7] for enhanced power control and hardware optimization methods is implemented using the Canonical Signed digit.

 Received on 10-10-2022, Revised on 11-12-2022, Accepted on 02-03-2023.

The (CSD) approach and the Hcub algorithm are utilized to calculate the size of the shifter. The Advanced Boolean Logic (ABL) technique used in the design of the adders blends the two binary adders into one, allowing for component sharing, especially during the preprocessing and sum computation phases. These three methods are utilized to restructure the complete shift and incorporate logic stages which [5], results in low power consumption and increased throughput.

The primary concept behind this suggested approach is that redundant arithmetic adders can be employed in place of a traditional CSLA adder. Due to no carry propagation chains; this adder may perform quick arithmetic operations in a variety of data processing approaches. Due to the aforementioned feature, this adder is mostly utilized for minimizing area and power dissipation. The carry propagation delay can be reduced by manipulating CSLA in various computational structures, but it cannot be eliminated entirely [13]. Compared to standard binary representation, a redundant binary representation is effective for accelerating arithmetic operations even on FPGA/ASIC. Redundant binary representation accelerates addition, subtraction, and multiplication. The price for the speed is frequently reasonable when you realize how crucial timing is when creating digital filters.

III. PROPOSED ALGORITHM

The shift-and-add algorithm is designed in hardware using the proposed method, which uses an integrated adder and subtractor to simplify the hardware implementation. Because it combines numerous adders/subtractors into a small one, the integrated adder/subtractor uses minimum adder, and subtractor for the computation of addition, subtraction, carry, and borrow. To calculate the nano-rotations, it makes use of some of the resources from the shift-and-add method and goes through multiple rotational steps. Each level in this series differentiates itself from the others based on the input angle for the shift-and-add nano-rotations. Instead, then using traditional logic gates, the proposed solution employs redundant arithmetic gates to produce the correct performance of addition and subtraction operations.

A. Integrator Adder and Subtractor

Using a shift-and-add algorithm, you can simplify long-term arithmetic operations by integrating the adder and subtractor. If more than two bits are added or subtracted, the binary full adder/subtractor will fail to work [8]. It can handle only one bit per input with output carry and borrow. A different kind of computer arithmetic that is appropriate for applications with a lot of numbers is provided by Redundant Number Systems (RNS). The ability of RNS is to catch or prohibit carry propagation [9], which results in parallel adders with constant delay regardless of the operand word length, is a key feature. As a result, RNS format results are produced with reduced latency. It uses 1.4 times less power and memory while reducing the number of adders and subtractors that are utilized in the shift-and-add architecture.

In numerically demanding applications, the usage of redundant number systems can greatly enhance computational performance. However, because each symbol requires numerous bits, the architecture of their arithmetic circuits is typically expensive [9]. The redundant arithmetic integrator has two types of circuits i.e., Redundant Arithmetic - Plus Plus Minus (RA-PPM) adder, and Redundant Arithmetic - Minus Minus Plus (RA-MMP) Subtractor. An integrated adder/subtractor consists of two RA-PPM and two RA-MMP gates with redundant arithmetic logic architecture as shown in Fig. 1.

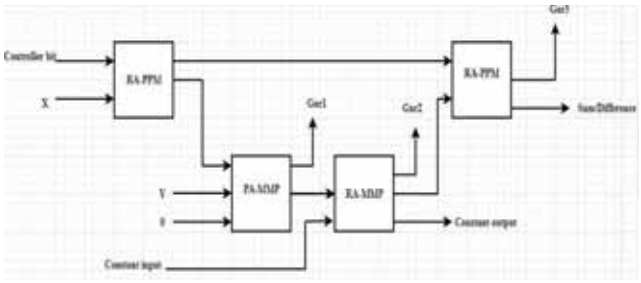

Figure 1. Redundant Arithmetic Integrator Adder & Subtractor

Three inputs—X, Y, and a constant input—are used, and the output is the sum or difference. When the control bit is set to zero, the circuit performs an addition operation. If the value is set to one, the circuit performs a subtraction operation. In this circuit, there is only one garbage input, which is represented by the logical zero. Gar1 through Gar3 refers to the three garbage outputs [10]. The third and the fourth inputs to the RA-MMP gate are the output from RA-PPM gates, constant inputs, and B value. Two split equivalent outputs are managed by A, the second input to the RA-PPM gate.

The selection input represents the constant input which is taken into consideration for addition and subtraction. The input indicates whether the operand is a 2'complement and whether it follows a critical path. Utilizing this shift-and-add architecture's combined adder and subtractor the complexity of addition and subtraction has been lowered, and the need for adders and subtractors was also decreased. In this study, the shift-and-add architecture [11, 12] is a simplified, fully integrated adder/subtractor, which increased architecture performance and reduced power consumption.

B. Improved Shift-and-Add Architecture

The improved shift-and-add algorithm's rotation stages are organized as shown in Fig.2 below, and Table 1 provides detailed information on each stage. The shift-and-add Rotator rotates the input vector by any necessary angle to produce a new vector with the x-axis. The result of the vectoring procedures and the original vectors' scaled magnitude is the rotation angle. When six stages of architecture is taken into consideration it may not be suitable for all kinds of applications because of slight latency increases which may not be majorly suitable for battery-operated devices. The six stages are Trival rotations, friend angle rotation, USR Cordic, traditional Cordic (2 stages), and Nano rotator. The Trivial Rotation stage measures the trivial rotations by ±180 and ±90 in the range of ±45 to set

the remaining angle apart from this any other angle set cannot be assigned. The friend angle rotation stage will rotate the entire kernel with a reduced input angle set. In the USR Cordic stage, the angle value will be further reduced by changing the shifter and reducing the number of iterations. A comparison of two angle sets which are coming from the friend angle and USR rotor will be compared and finally, that will be given to the nano rotation stage. The output of this stage will have a very minute angle value. The CORDIC algorithm, where it is bigger than out of the preceding stage for every stage, provides the convergence of circumference. Finally, the presentation of the angle in the range of [0, 1]. By eliminating the 4th and 5th stages angle minimization is happening but latency is slightly increasing due to its traditional architectural structure of Cordic which is designed using carry save adders and no. of iterations. When iterations are going on increasing the area of architecture is slightly increasing due to an increase in adders and shifters. The main goal of architecture is to minimize the angle and to get in range of [0, 1] so the 4th and the 5th stages are eliminated and slightly modify the architecture of the nano-rotor with different shifter sizes, and a few multiplexers are added.

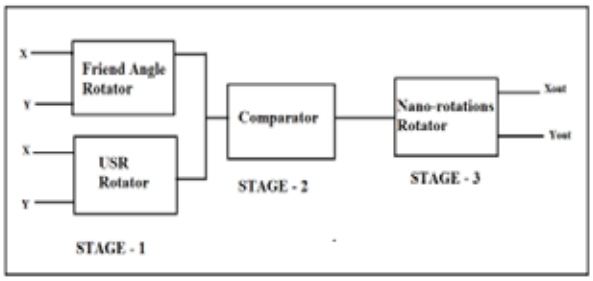

Figure 2. The architecture of the Improved Shift-and-Add Algorithm

TABLE I.
ROTATION STAGES AND THEIR DESCRIPTION

Stage Index	Input Angle	Output Angle
1	±180°	±46°
2	±48.105	±11.305
3	±11.67	±4.635
4	±4.56	±2.79
5	±2.79	±0.986

At each rotation of the angle, the function of the vectoring uses the least significant components of the residual vector. The sign of the residual component determines the next rotational direction of the rotator. The angle accumulator is first initialized to zero, and the traversed angle specifies the end of each repetition.

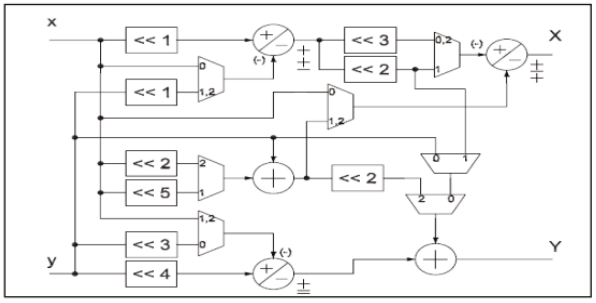

Figure 3. Architecture of Friend Angle Rotator

The canonical signed digit, which subdivides the original angle into several smaller angles, is used to extract the rotational quantization process of the angle. These sub-angles reduced the angle quantization error and are roughly closer than that of the original rotational angle. In the initial phase, friend angle and USR [11] are used. The hardware architecture is presented in Fig. 3. It has seven 2:1 multiplexers and five adders. Depending on how the multiplexers are set up, it can calculate all the kernel rotations. The kernel is also defined by the user. The Rotator architecture of the USR is depicted in Fig. 4. The only way to implement this architecture is by using two adders and two 2:1 multiplexers.

The friend angle and USR inputs in STAGE-1 are both presented in rectangular form. The outputs of the Friend angle rotator and USR rotator are compared using two 16-bit comparators as shown in Fig. 5 in STAGE-2. Separate comparisons are made between the Friend angle rotator's output x out and y out and the USR.

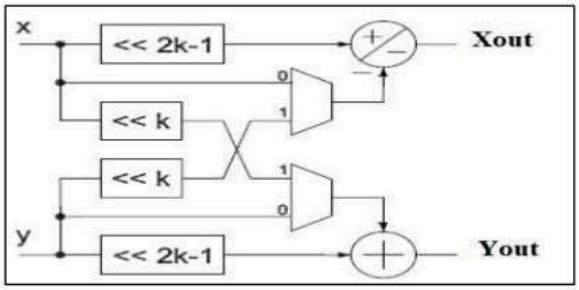

Figure 4. The architecture of the USR Rotator

The friend angle and USR inputs in STAGE-1 are both presented in rectangular form. The outputs of the Friend angle rotator and USR rotator are compared using two 16-bit comparators as shown in Fig. 5 in STAGE-2. Separate comparisons are made between the Friend angle rotator's output x out and y out and the USR.

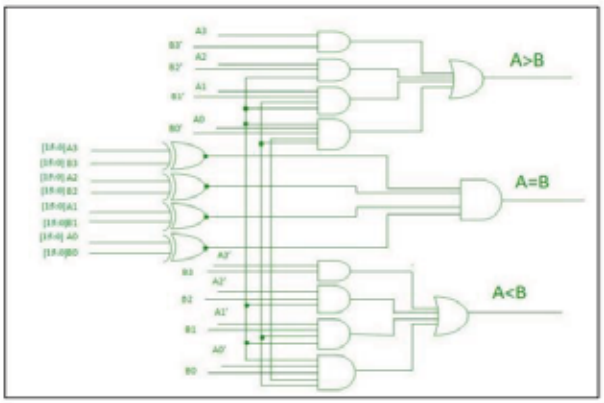

Figure 5. Design Structure of 16-bit Comparator

This level employs a rotator with nano-rotations as shown in Fig. 6. First, β_M must be chosen to construct the rotator taking into account the range of input angles.

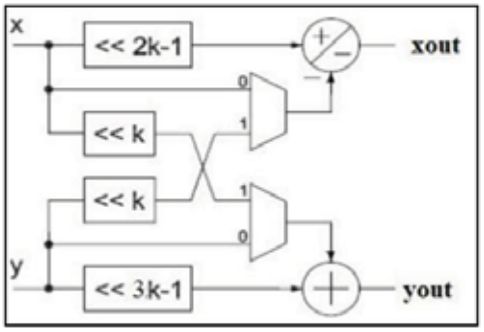

Figure 6. Architecture of Nano-Rotations

The output of the comparator is used as input to the Nano rotator. Depending on the situation, the user can select either the value A>B or AB. The user can use AB if he only wants a very small angle. The output is also reliant on the 2:1 multiplexer's selection line.

IV. Experimental Results & Discussion

The Carry Select Adder (CSLA), which mandates that the carrying extend from the LSB to the MSB, is the foundation of the improved Shift-and-Add architecture. As a result, as word length rises, so do the flaws in the breakdown process and the execution of the sub functions [12]. Because of this, there are various critical routes for various stages, which cause distinct clock cycles. The core's general working frequency is therefore determined by the system's slowest arrangement, which is a factor. The proposed redundant arithmetic adder (RA-PPM) is significantly less complicated, requiring only bit-by-bit XOR, NAND, and INV as shown in Fig. 7, as well as synchronous activities to make the limitation in the disintegration of a level planning reaction and primarily reducing long carry prorogation chains.

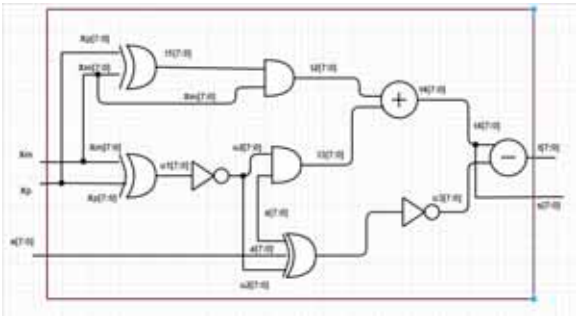

Figure 7. RTL Schematic of RA-PPM adder

The CSLA logic is created with the use of parallel carry chains. The suggested RA-PPM adder structure [12] and traditional structures, such as the CSLA [12], are compared in terms of various area execution parameters, including the number of cells, I/O ports, and nets in addition to a logic latency (ns), path latency (ns), and highest combinational delay (ns) shown in Fig. 8. In Fig. 9, the power usage of two designs is displayed. Compared to the CSLA structure, the suggested RA-PPM structure dissipates less power (Fig. 9). A superfluous mathematical idea diminished the strength of the suggested method. In addition, the suggested Shift-and-Add use little hardware, leading to increased on-chip control dispersion. It is obvious that the planned Shift-and-Add is practical to enforce when it comes to the planning reaction of the suggested Shift-and-Add structure.

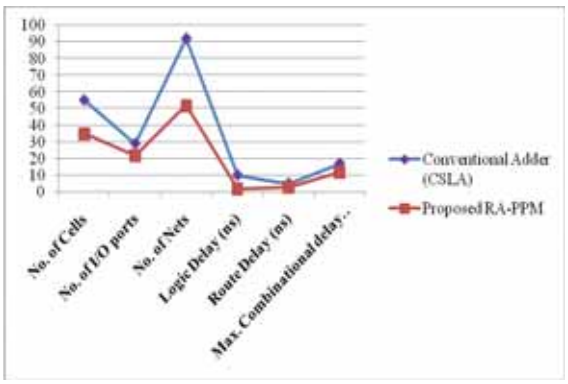

Figure 8. Comparison of CSLA and RA-PPM in terms of area

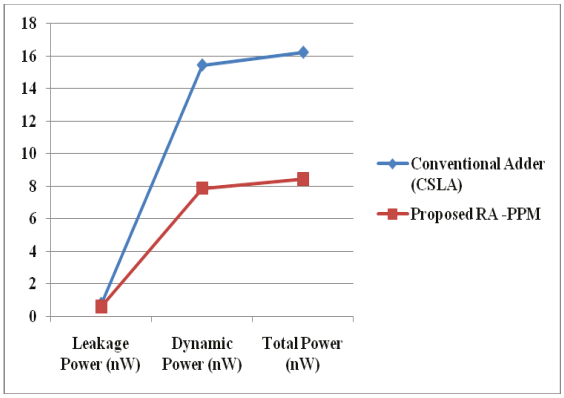

Figure 9. Comparison of CSLA and RA-PPM in terms of power

RTL schematic of the RA-MMP subtractor is shown in Fig. 10.

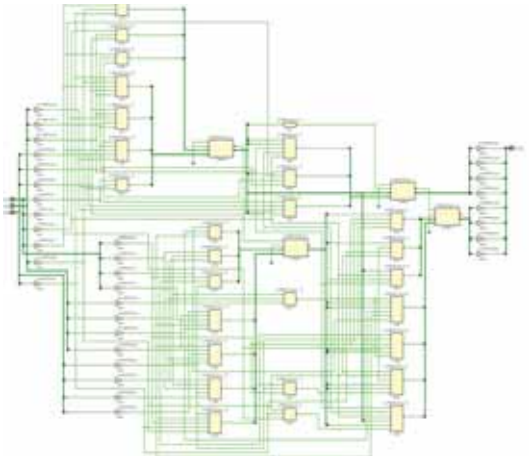

Figure 10. RTL Schematic of RA-MMP subtractor

The rectangular shape of the input angle is taken. Specifically, x+ iy = 25+ i500 = 85.120 (angle). In stage 1, the input is provided to the buddy angle rotator and USR. The 2:1 MUX selection line of the rotators control output of stage 1. Then, in stage 2, the comparator compares the outputs of the two rotators, and the lowest value of x + iy is

CVR Journal of Science and Technology, Volume 24, June 2023
DOI: 10.32377/cvrjst2404

taken into account. The nano-rotation rotator receives the lowest value in stage 3.

CASE 1:
INPUT: x+ iy = 25+ i500 = 85.12° (angle)
Selection lines of 2:1 MUX of all rotators = s = 1
OUTPUT: Stage -1: Friend angle: 625+ i10425
USR CORDIC: 3175+ i64500
Stage-2: compare 625, 3175 and 10425, 64500.
Stage-3: lowest value of comparator is input to nano-rotator.
x+ iy = 625+ i10425
Finally, the output of the nano-rotation rotator is.
x+ iy = 49551 + i3257 = 3.76° (angle)

CASE 2:
INPUT: x+ iy = 25+ i500 = 85.12° (angle)
Selection lines of 2:1 MUX of all rotators = s = 0
OUTPUT: Stage -1: Friend angle: 5500+ i16200
USR CORDIC: 60736+ i64400
Stage-2: compare 5500, 60736, and 16200, 64400.
Stage-3: lowest value obtained by the comparator is the input to the nano-rotation rotator
x+ iy = 5500+ i16200
Finally, the output of the nano-rotation rotator is.
x+ iy = 19456 + i57344 = 71° (angle). The RTL schematic of redundant integrator adder and subtractor-based improved shift-and-add architecture is shown in Fig. 11.

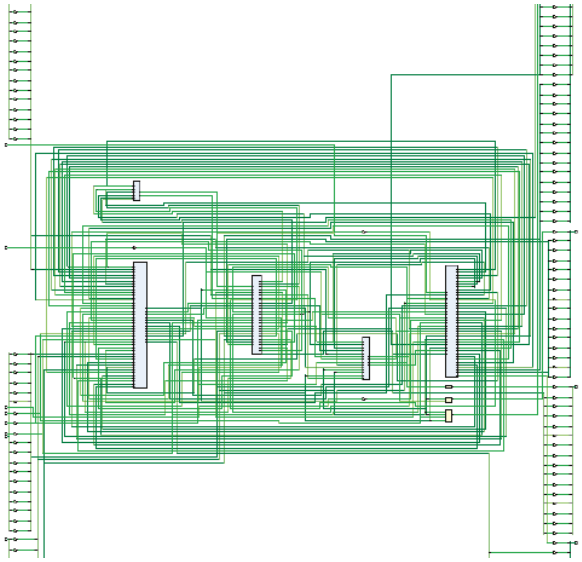

Figure 11. RTL Schematic of Improved Shift-and-Add architecture.

As it is designed using the suggested RA-PPM adder, it delivers lower hardware utilization and power consumption when compared to typical architectures. The input signal "Clk," along with all other input signals, is rotated at a 45-degree angle as one example. Sine values are discrete, while cosine values are discrete. The addresses are kept in predetermined Shift-and-Add algorithm slant values. Shift-and-Add architecture with the suggested adder is laid out in Fig. 12.

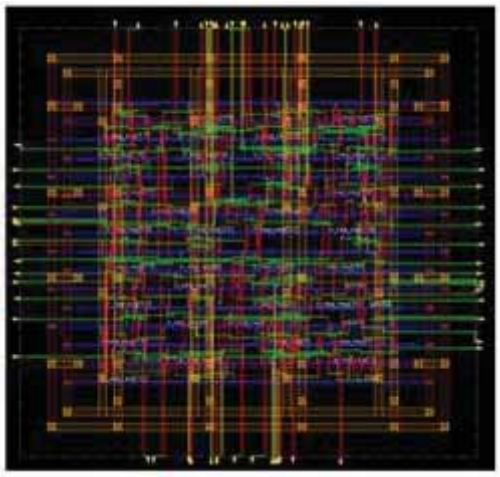

Figure 12. Physical Layout of Improved Shift-and-Add architecture.

Digital signal processing techniques typically call for a lot of multiplications. Consider a FIR filter with 32 taps as a basic example. A single sample of the filter output requires 16 multiplications, which becomes easy when it uses symmetry in the filter coefficients. These multiplications may take a while, depending on the number of bits used to represent the input samples and the filter coefficients. Over the years, there has been a lot of interest in the design of hardware-efficient multipliers, and adders, and substantial research has produced several solutions. The fundamental ideas of canonical signed digit (CSD) representation are reviewed in this article. When multiplying an input signal by a constant multiplicand, such as when designing an FIR filter where we have some fixed coefficients, CSD is an intriguing method for creating efficient multipliers. The arithmetic procedure known as Multiple Constant Multiplications (MCM) multiplies a group of fixed-point constants by the same fixed-point variable X. Costly multipliers must be avoided for MCM to be applied effectively. Hardware substitutions that solely do adds, subtraction, and shifts are required to be multiplier-less. As a result, the process of determining the least amount of addition/subtraction operations is how the MCM problem is described. It is hypothesized that MCM's computational complexity is NP-hard. Table II bounds have been precisely determined by the average, added cost, and adder depth additions for various algorithms. Compared to all algorithms CSD looks better for both serial and parallel adder design depths.

TABLE II.
COMPARISON OF VARIOUS ALGORITHMS IN TERMS OF ADDER COST

Algorithm	Adder-Cost	Serial-Adder Depth		Parallel-Adder Depth	
		Max.	Avg.	Max.	Avg.
CSD	12	10	8.33	4	4.00
HCUB	13	11	7.33		
MCM	19	8	6.33	4	4.00

V. CONCLUSIONS

In this study, a hardware reduction technique-enhanced shift-and-add design with an integrated adder and subtractor using a redundant arithmetic concept is proposed. An integrated adder/subtractor, or CSD approach, reduces the number of adders, subtractors, and shifters and by using RA-PPM adders and RA-MMP subtractors the latency is reduced, and throughput increases due to no carry propagation chain. ASIC-Cadence tools are used to implement the proposed and enhanced shift-and-add architecture. When compared to the existing methodologies, the performance of the suggested method performs better in terms of area, power consumption, and frequency. All the designs are designed using Verilog HDL, simulated using the NCHDL simulator, and synthesized using the genus synthesis tool, and finally, physical design is performed using the Cadence Innovus tool to generate a graphic data stream II file. When compared to the shift-and-add algorithm's conventional way, the simulation results reveal that the proposed method uses 12.11% less power, operates at a higher frequency of 33.3%, and occupies 14% less space.

REFERENCES

[1] J. E. Volder, “The CORDIC trigonometric computing technique,” IRE Transactions on Electronic Computers, vol. 8, no. 3, pp. 330–334, 1959.

[2] J. S. Walther, “A unified algorithm for elementary functions,” in Proceedings of the AFIPS Spring Joint Computer Conference, pp. 379–385, May 1971.

[3] P. K. Meher, J. Valls, T. Juang, K. Sridharan, and K. Maharatna, "50 Years of CORDIC: Algorithms, Architectures, and Applications," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 56, no. 9, pp. 1893-1907, Sept. 2009.

[4] Parfieniuk, Marek, and Sang Yoon Park. "Sparse-iteration 4D CORDIC algorithms for multiplying quaternions." IEEE Transactions on Computers 65.9 (2015): 2859-2871.

[5] Mario Garrido, Petter Källström, Martin Kumm, and Oscar Gustafsson, “CORDIC II: A New Improved CORDIC Algorithm”, IEEE Transactions on Circuits and Systems—II: Express Briefs, vol. 63, no. 2, February 2016.

[6] Sharath Chandra Inguva, Dr. J. B. Seventline, “Implementation of Area Efficient CORDIC for QPSK Modulation”, Journal of Engineering Science and Technological Review, Volume 11, Issue 2, Pg:96- 102,2018.

[7] Sharath Chandra Inguva, Dr. Joseph Beatrice Seventiline. "LHCORDIC: Low Power FPGA Based Implementation of CORDIC Architecture", International Journal of Intelligent Engineering and Systems, Vol.12, No.2, 2019.

[8] Madhavi, Hossein, and Somayeh Timarchi. "Area-Time-Power Efficient FFT Architectures Based on Binary-Signed-Digit CORDIC." IEEE Transactions on Circuits and Systems I: Regular Papers (2019).

[9] Kamp W., Smith A. B., Hayes M.,‖ Efficient Implementation of Fast Redundant Number Adders for Long Word-lengths in FPGAs,‖ International Conference on Field-Programmable Technology, 2009.

[10] Nupur Jain, Biswajit Mishra, and Peter Wilson, "A Low gate count Reconfigurable architecture for biomedical signal processing applications", SN Applied Sciences (A Springer Nature Journal), 3:439, 2021.

[11] Adam Wilson J, Gerwin Schalk, and Justin Williams, "A Procedure for Measuring Latencies in Brain-Computer Interfaces, IEEE Transactions on Biomedical Engineering", 57(7), pp: 1785 - 1797, 2010.

[12] Subha Sri Lakshmi. Thiruveedhi, P. Radhika Yedukondalu Kamatham," Performance Analysis of Shift-And-Add Architecture Using RBA Adder for Low Latency Biomedical Signal Processing Applications ", 2nd International Conference on Intelligent Technologies (2nd CONIT 2022) Technical Co-sponsored by IEEE Bangalore Section, K. L. E Institute of Technology, Hubballi, Karnataka, India, 24th – 26th June 2022.

[13] Mr. Sharath Chandra Inguva, Dr. J. B. Seventline, “Enhanced CORDIC Algorithm using an Area Efficient Carry Select Adder”, International Conference on Intelligent Sustainable Systems (ICISS 2017), 978-1-5386-1959-9/17/$31.00, 2017 IEEE.