

Binary Vedic Multiplication Using Carry Save Adder CS3A based on Modified Approximate Three Operand Adder

A. Priyanga¹, C.N. Marimuthu²

¹PG Scholar, Department of Electronics and Communication Engineering, Nandha Engineering College.

²- Professor, Department of Electronics and Communication Engineering, Nandha Engineering College.

Abstract- Consumer electronics markets have increased demand for high-speed, low-power adders with huge operands to be used in new portable systems. One of the most promising ways for achieving a trade-off between delay and power consumption for the addition of big operands is the CSA adder. In this paper, a VLSI architecture is proposed based on Binary Vedic multiplication using Carry save adder. The suggested changing binary Vedic multiplication technique is more efficient in terms of delay. The Vedic multiplication method can be extended for a larger bit size. The Xilinx ISE Design Suite 14.2 is used for circuit synthesis. The simulation results for 4 bit and 8 bit multiplication. VLSI implementation results when compared with existing it reveal that the suggested adder saves more than 12% of energy and reduces the area-delay-product by more than 5%.

Index Terms— Three-operand adder, carry save adder (CSA), Vedic Multiplication.

I. INTRODUCTION:

The word “Vedic” is derived from the word “Veda” which means the store-house of all knowledge. The word “Vedic” comes from the word “Veda” which implies the store-house of all information Vedic arithmetic is associate ancient system of arithmetic existed in Asian nation. Vedic arithmetic is far less complicated and simple to know than typical mathematic. Developing digital systems is hampered by the number. The spread of strategies for implementing quick multipliers within the literature. Another approach for implementing an efficient number is that the Vedic multiplication formula. In Vedic arithmetic, there are three ways to travel concerning multiplying. nVedic of the 3 strategies is generic, that means it's going to be employed in any situation; the opposite are situation-specific. Urdhva Tiryakbhyam is that

the primary vedic multiplication formula. Literally, it signifies in 2 directions: up and down.

This multiplier number multiplies 2 operands vertically and crosswise then, adds the ensuing sums.

$$\begin{array}{r}
 4 \ 6 \\
 \times 3 \ 3 \\
 \hline
 1 \ 8 \\
 1 \ 2 \times \\
 1 \ 8 \times \\
 1 \ 2 \times \times \\
 \hline
 1 \ 5 \ 1 \ 8
 \end{array}
 \begin{array}{l}
 \leftarrow 3 \times 6 \\
 \leftarrow 3 \times 4 \\
 \leftarrow 3 \times 6 \\
 \leftarrow 3 \times 4
 \end{array}$$

Fig 1: 2x2 Vedic Multiplication

Multipliers are key parts of the many high performance systems such as FIR filters, Microprocessors, Digital Signal Processors etc. To perform multiplications, an oversized range of adders or parts are used. The traditional system of arithmetic named as vedic multiplier was rediscovered from the Vedas. In distinction to traditional method, this vedic multiplier is easier and simple to know.

It includes sixteen-sutras or formulae and 13 sub-sutras. The variability of applications of this multiplier includes theory of numbers, compound multiplication, pure mathematics operation, calculus, squaring, cubing, cube root, easy quadratic, geometry and marvellous to vedic Numeric Code. The speed is a vital consider the third-dimensional VLSI drawback and conjointly a constraint within the multiplication operation. Therefore increase in speed will be achieved by sinking the quantity of steps within the computation method. Therefore the potency of the system will be evaluated by the assistance of Speed and space consumed by the parts of number determines the potency of a system.

The technique use in this multiplier is principally supported sixteen Sutras. Vedic multiplier techniques

would end in the saving of process time and underneath the sixteen Sanskrit literature principally used “Urdhva Tiryakbhyam”.

II. LITERATURE SURVEY

Digital Signal Processing (DSP) is a very significant and active research area. High throughput is a requirement for most wireless communication systems. The critical bottleneck that affects communication ability is the Fast Fourier Transform (FFT), which is the essence of most modulators. Currently, Floating point FFT processors have been used in Radar signal processing, fast convolution, Spectrum estimation and OFDM based modulators/demodulators. Efficient VLSI based architectures are required for real-time FFT processing. The multiplication limits the performance in terms of throughput of FFT. Consequently, there is a need for high speed and low power multiplier architectures with minor truncation error. The present paper presents a modified binary floating-point multiplier using Vedic mathematics and a modification in the previously published Vedic multiplier circuit has been proposed. The entire design has been implemented in Verilog HDL. [1]. Multipliers act as processors and take on the notable work of many computing frameworks. The speed of the processor is profoundly affected by the speed of their multipliers. In order to improve the system speed, faster and more efficient multipliers should be used. A Vedic multiplier is one of the best solution that can be used to perform multiplications at a faster rate by eliminating the steps that are not needed in usual multiplication process. Power consumption is another critical issue in embedded systems that cannot be ignored. Reversible logic has become notable in the recent years because of its potential to reduce power utilization, which is a major concern in digital design. In this work, a high-speed 16×16 Vedic multiplier was designed using Urdhva Tiryagbhyam (UT) sutra, which is derived from Vedic mathematics. [2]. Digital Signal Processors play an unavoidable portion in modern-day communication. The Multiply Accumulate (MAC) is a crucial component of modern signal processors that performance is rely on speed, power, and area. In this paper, one such

promising option is the Vedic multiplier based on the modified sum-product method is proposed [3]. The main goal of this paper is to design an efficient 2D FIR digital filter for digital image processing and digital signal processing applications. To optimize filter speed, area and power different multipliers like array, Wallace tree, Booth and Vedic are used in the design of filters. Among these multipliers, Vedic multiplier reduces the partial products in multiplication. This increases the speed of the multiplication process. Vedic multiplier is based on ancient mathematics and uses a sutra called “Urdhva Tiryabhyam”. This paper, proposes two methods to optimize speed, area and power. [4]. Fast Fourier Transform is one of the most efficient methods of performing computation in Digital signal processing blocks. These computations are basically performed by the inherent floating-point multiplier units residing inside the butterfly units of any FFT. To optimise an FFT for higher efficiency and performance, it is inevitable to use highly efficient adder and multiplier units within the datapath architecture of a FFT processor. This work proposes high-performance FFT units designed using optimised Vedic multiplier units for DSP processor cores. Choice of Vedic multiplier decreases power and delay overheads of design which leads to production of an efficient FFT. A comparative analysis is presented for 24-bit Vedic multiplier using nine adder variants [5]. In this paper, the approach to generate PN sequence using Vedic multiplier has been proposed. In a Vedic multiplier, multiplication is done by using Urdhva Tiryagbhyam sutra, and basic multiplication method is used to design conventional multiplier. The PN sequence generator is implemented using Verilog HDL. The comparative analysis of Vedic multiplier over the conventional multiplier is implemented using Altera FPGA. Vedic multiplication gives an effective result when compared to conventional multipliers. [6]. Addition, Subtraction and Multiplication are commonly used basic arithmetic operations. In these operations, Multiplication requires more processing time than other Arithmetic operations. In this paper, we will discuss various multiplication techniques. For fast multiplication, here we are using Vedic multiplication technique, which is one of the most popular

multiplication technique. In this paper, comparative study of types of Vedic Multiplier and Conventional Multiplier is done. We have designed a 16-bit Multiplier using Vedic Multiplication technique (Urdhva Tiryagbhyam sutra, Nikhilam Sutra and Ekadhikena Purvena Sutra) and a conventional multiplier for generating the partial products. [7]. Optimising multiplier (delay or area or power) will have a huge impact on system performance. One of the methods adopted to reduce delay is the use of Vedic Multiplier based on "Vedic Mathematics Sutras". In this project "Urdhava - Tiryakbhyam" Sutra is used to perform multiplication and is simulated and implemented in Xilinx ISE Design 14.7[8]. The FFT Butterfly Structure work is designed with a "Vedic Multipliers" for applications at high speed. In this Vedic Multiplier, an algorithm called "Urdhva Tiryabhyam" was used to improve its efficiency by optimizing the number of logic gates, constant inputs and garbage outputs. The Data Computation time is reduced by an 3-1-1-2 compressor using reversible logic gates. Hence reducing the surplus power consumption of 11.24% and summation of the partial products is done with less delay factor of about 5.28%. The area, power, delay, area delay product and power delay product are calculated using cadence virtuoso and is implemented in Spartan-6 device family using Xilinx ISE [9]. In this Paper, an efficient implementation of the direct form of the least mean square (LMS) adaptive filter algorithm is proposed. The conventional multiplier hinders the speed of the delayed LMS (DLMS) algorithm; hence latest high speed Vedic Multiplier is used for its high convergence rate, further, the Vedic Multiplier is explored for reducing the number of logic levels and timing levels, and possible reduction in logic delay. The efficient adders are used in digital signal processing applications to reduce the power requirement, area and delay [10].

III. THREE OPERAND ADDER BINARY TECHNIQUES:

The three-operand binary addition is one among the important operation within the congruential standard arithmetic architectures and LCG-based PRBG ways like CLCG, MDCLCG and CVLCG. It is enforced either by mistreatment 2 stages of two-operand adders or one stage of three-operand adder. Carry-save adder (CSA) is that the usually used technique to perform the three-operand binary addition. It computes the addition of 3 operands in 2 stages. The primary stage is that the array of full adders. Every full adder computes "carry" bit and "sum" bit at the same time from 3 binary input a_i , b_i and c_i . The second stage is that the ripple-carry adder that computes the ultimate n-bit size "sum" and one-bit size "carry-out" signals at the output of three-operand addition. The "carry-out" signal is propagated through the n variety of full adders within the ripple-carry stage.

Therefore, the delay will increase linearly with the rise of bit length. The design of the three-operand carry-save adder is shown in Fig. one and therefore the important path delay is highlighted with a broken line. It shows that the important path delay depends on the carry propagation delay of ripple carry stage and is evaluated as follows,

$$TCS3A = (n+1)TFA = 3TX + 2nTG$$

Similarly, the overall space is evaluated as follows,

$$ACS3A = 2nAFA = 4nAX + 6nAG$$

Here, noble metal and TG indicate the world and propagation delay of basic 2-input gate (AND/OR/NAND/NOR) severally.

IV. EXISTING SYSTEM

In this existing technique, 3 operand adder technique is employed. This adder circuit comprises 5 logic circuit. That area unit Bit-addition logic, Base logic, PG logic (Black cell and gray cell), and besides that add logic. Add logic circuit is that the ending of the 3 quantity adder circuit. The input of the bit addition logic circuit could be a,b,c then the output is add and carry. Then the output of add and carry is given to the input of next base logic circuit. In Base logic propogation and generation are generated. when the PG logic the ultimate add log-

ic are dead. By mistreatment Xilinx ISE fourteen. The result are generated.

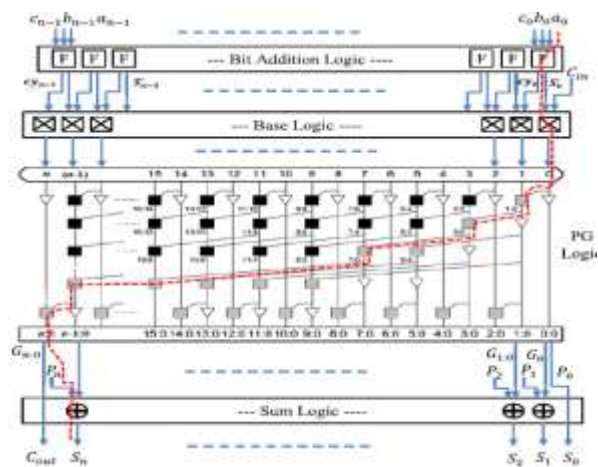


Fig2: Block diagram of Existing Method

The logical expression of of these four stages area unit outlined as follows,

Stage-1: Bit Addition Logic

$$S_i = a_i \oplus b_i \oplus c_i$$

$$c_{i+1} = a_i \cdot b_i + b_i \cdot c_i + c_i \cdot a_i$$

Stage-2: Base Logic

$$G_{i:i} = G_i = S_i \cdot c_{i-1}, G_{0:0} = G_0 = S_0 \cdot C_{in}$$

$$P_{i:i} = P_i = S_i \oplus c_{i-1}, P_{0:0} = P_0 = S_0 \oplus C_{in}$$

Stage-3: PG (Generate and Propagate) Logic

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

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

Stage-4: add Logic

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

V. PROPOSED SYSTEM

The hardware security within the field of IoT applications demands stream-cipher based mostly high rate, light-weight cryptography technique for quickest encryption/ decoding. Key generator or pseudorandom bit generator (PRBG) is that the primary element within the stream-cipher based mostly encryption/ decoding. changed dual-CLCG (MDCLCG) is that the most effective PRBG tech-

nique amongst the prevailing PRBG ways that is appropriate for stream-cipher based mostly hardware security. However, the protection strength of the MDCLCG technique linearly depends on the bit size of the congruential modulus. it's polynomial-time unpredictable and secure if $n \geq 32$ - bits.

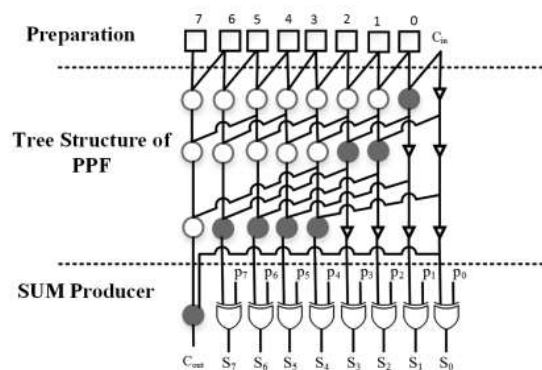


Fig3: Proposed System block diagram

In proposed system, the first stage of a PPF adder, named the preparation unit, produces generate (g), alive (a), and propagate (p) signals. Let $X = x_n - 1x_n - 2 \dots x_0$ and $Y = y_n - 1y_n - 2 \dots y_0$ be the two operands. The g, a, and p signals for each bit position can be calculated as follows :

$$g_i = x_i \cdot y_i$$

$$a_i = x_i + y_i$$

$$p_i = x_i \oplus y_i$$

In order to improve the area, the proposed design adopts two kinds of blocks for computing the sum. The Sum Producer type 2 is a sort of carry select adder with a different structure. It is known that the delay of CSL adder is less than other adders. The MDCLCG design given in [10] is developed with four three-operand modulo-2n carry-save adders (CS3A) and 2 magnitude comparators together with four registers and multiplexers. The longer carry propagation gate delay in CS3A adder influences the performance of MDCLCG design with associate degree increase of bit size. Therefore, in this section, the performance metrics of the MDCLCG square measure measured by replace-

ment the CS3A adder with the HHC3A and planned adder architectures.

By considering the operation of three-operand modulo-2n addition in MDCLCG technique, the design of the planned adder is additional redesigned. Therefore, the area (AP3OA) and time (TP3OA) quality of the planned adder design are often evaluated for three-operand modulo-2n addition as follows,

$$TP3OA \approx 4TX + \text{two } \log_2 nr + \text{one } TG$$

Here, $s \log_2 n$ one and nr n one. Similarly, the area (Amgc) and time (Tmgc) quality of the magnitude

comparator is evaluated in [10] that is additional highlighted as follows,

$$Amgc \approx (n-1) [9AG + 4AN]$$

$$Tmgc \approx \text{four}TX + 4 \log_2 n TG$$

Therefore, the area and time quality of the MDCLCG design victimization the planned adder and therefore the magnitude comparator are often evaluated as follows,

$$TMDCLCG \approx T3oa + Tmx$$

$$4TX \ 2 \log \ nr \ 2 \ TG$$

$$AMDCLCG = \text{four } Amx + A3oa + Arg + 2Acmp + AX$$

$$\approx (16n - 7) AX + +4 (3n - 2) \text{ associate degree} + 4nAFFg$$

It is ascertained that the planned adder is quicker than the CS3A, HC3A and HHC3A primarily based MDCLCG architectures. Similarly, it consumes sixteen.5% less space and fifteen.1% less power than the HC3A- primarily based MDCLCG design. Moreover, it achieves all-time low ADP and PDP than the CS3A and HC3A primarily based MDCLCG architectures. It's reportable the reduction of forty eight.7% and 36.2% ADP as compared to the CS3A and HC3A primarily based MDCLCG architectures.

VI. VEDIC MULTIPLIER:

2X2 Vedic Multiplier

The basic 2x2 binary multiplier circuit using two half adder (HA) modules. 2-bit inputs are A = A1A0 and B = B1B0 and R2R1R0 are output bits.

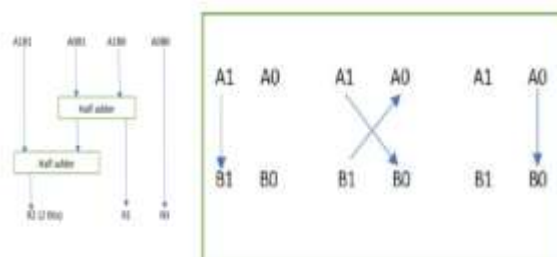


Fig 3: Method of 2x2 multiplier

Step 1: Take right most digit multiple vertically i.e A0B0.

Step 2: Take all two digits multiple diagonally i.e A1B0+A0B1.

Step 3: Take left most digits multiple vertically i.e A1B1.

4X4 Vedic Multiplier

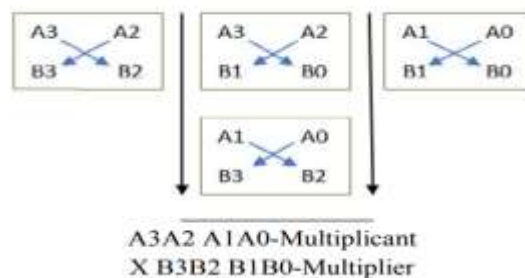


Fig4: Method of 4x4 multiplier

Above each block shows the 2 bit multiplier. A1A0,B1B0, are inputs for first 2 bit multiplier Next the middle blocks shows the 2x2 multiplier with inputs "A3A2" & "B1B0" and "A1A0" & "B3B2" Last block shows the 2x2 multiplier with inputs "A3 A2" and "B3 B2" So it will get final output as 8 bit. i.e. is "S7S6S5S4S3S2S1S0".

8x8 Vedic Multiplier

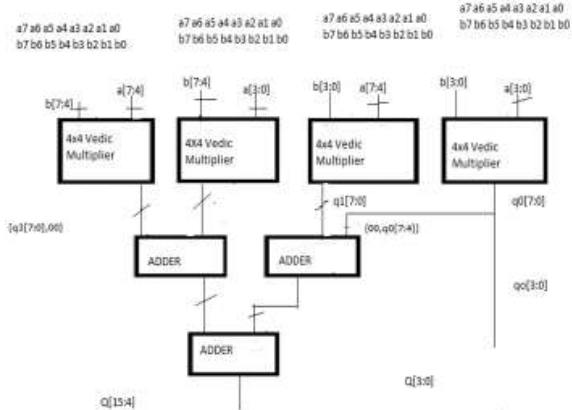


Fig5: Block diagram of 8*8 bit multiplier

Comparison with Various Architectures The 8-bit multiplier designed is compared with various architectures in terms of total delay, logic delay, route delay and number of logic levels. The results obtained are tabulated in Table I. From table I, it is evident that there is a reduction in both total delay and logic levels. The routing delay is found to be 9.424 ns, the logic delay is 5.626 ns; thus, giving a total delay of 15.050 ns. The number of logic levels is 11. Thus, it is clear that the proposed design outperforms the other popular multiplier architectures. The proposed architecture can be used to develop a high speed complex number multiplier with reduced delay.

VII. RESULT AND DISCUSSION

The proposed 8-bit multiplier is coded in Verilog HDL, simulated using Xilinx ISim simulator, synthesized using Xilinx XST for Spartan 6: xc6slx4-3tqg144 FPGA and verified for possible inputs given below. Inputs are generated using Verilog HDL test bench. The simulation result for 8-bit multiplier is shown in the Figure 9.

CASE - 1: Inputs a = “11111111”, b = “11101101”
Product p = “1110110000010011”

CASE - 2: Inputs a = “10001001”, b = “01001001”
Product p = “0010011100010001”

CASE - 3: Inputs a = “01010110”, b = “01000000”
Product p= “0001010110000000”

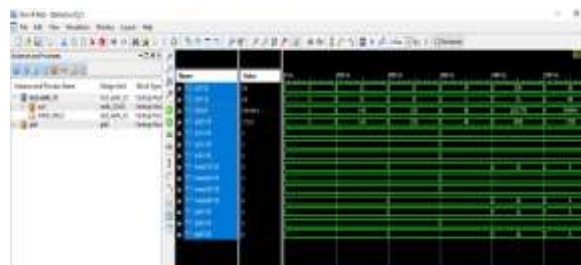


Fig 6: Output Waveform for 8*8 bit multiplier

Comparison with Various Architectures The 8-bit multiplier designed is compared with various architectures in terms of total delay, logic delay, route delay and number of logic levels. The results obtained are tabulated in Table I. From table I, it is evident that there is a reduction in both total delay and logic levels. The routing delay is found to be 9.424 ns, the logic delay is 5.626 ns; thus, giving a total delay of 15.050 ns. The number of logic levels is 11. Thus, it is clear that the proposed design outperforms the other popular multiplier architectures. The proposed architecture can be used to develop a high speed complex number multiplier with reduced delay.

Table 1: Results comparison for Vedic Multiplier

Performance Analysis			
S.No	Parameter	Existing Method	Proposed Method
1	Luts	141	116
2.	FlipFlops	48	25
3	Power in W	0.151	0.64
4.	Combinational Delay in ns	15.595	17.036

From table I, it is evident that there is a reduction in both total delay and logic levels. The routing delay is found to be 9.424 ns, the logic delay is 5.626 ns; thus, giving a total delay of 15.050 ns. The number of logic levels is 11. Thus, it is clear that the proposed design outperforms the other popular multiplier architectures. The proposed architecture can be used to develop a high speed complex number multiplier with reduced delay.

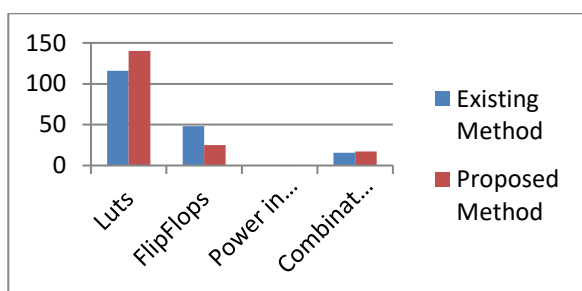


Fig7: Comparison chart of Existing and Proposed Method

VIII CONCLUSION:

An efficient novel technique is presented for binary multiplier circuits based on Vedic mathematics. The goal of this project is to design and implement and analysis of Vedic Multiplier architectures based on Urdhva Tiryakbhyam sutra in Vedic Mathematics. It is proved from the synthesis results that the proposed technique is much efficient in terms of delay. The proposed technique can be extended for a larger bit size. The power analysis has also been done in this work. Reducing the delay is an important requirement for various applications using Carry Save Adder and Vedic Multiplication technique is appropriate for this purpose.

REFERENCES:

[1] Amit Kumar Panda, Rakesh Palisetty & Kailash Chandra Ray, "High Speed Area Ef-

ficient VLSI Architecture of Three Operand Binary Adder", IEEE Trans.Vol.67 No.11, Nov 2020.

- [2] A. Kumar Panda and K. Chandra Ray, "A coupled variable input LCG technique and its VLSI architecture for pseudorandom bit generation," IEEE Trans. Instrum. Meas., vol. 69, Apr. 2020.
- [3] Xifan Tang, Edouard Giacomini, Giovanni De Michelli, "FPGA-SPICE: A Simulation based Architecture Evaluation based Framework for FPGAs", IEEE Trans.Vol.27 No.3, Mar 2019.
- [4] A. K. Panda and K. C. Ray, "Design and FPGA prototype of 1024-bit Blum-Blum-Shub PRBG design," in Proc. IEEE Int. Conf. Inf. Commun. Signal Process. (ICICSP), Singapore, Sep. 2018, pp. 38-43.
- [5] S. S. Erdem, T. Yanik, and A. Celebi, "A general digit-serial architecture for montgomery standard multiplication," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 25, no. 5, pp. 1658-1668, May 2017.
- [6] Shh-Lun Chen, Min-Chun Tuan, Ho-Yin Lee and Ting-Lan Lin, "VLSI Implementation of a Cost-Efficient Micro Control Unit With an Assymmetric Encryption for Wireless Body Sensor Networks", IEEE Trans.Vol.5, Apr 2017.
- [7] Shh-Lun Chen, Min-Chun Tuan, Tse-Yen-Liu, Chia-Wei-Shen, "VLSI Implementation of a Cost-Efficient Near-lossless CFA Image Compressor for Wireless capsule Endoscopy", IEEE Trans.Vol.4, Jan 2017.
- [8] S. Muthyala Sudhakar, K. P. Chidambaram, and E. E. Swartzlander, "Hybrid Han-Carlson adder," in Proc. IEEE 55th Int. Midwest Symp. Circuits Syst. (MWSCAS), Boise, ID, USA, Aug. 2012
- [9] D. L. Harris, "Parallel prefix networks that create trade offs between logic levels, fanout and wiring racks," Nov. 11, 2004.
- [10] B. Parhami, Computer Arithmetic: Algorithms and Hardware Design. New York, NY, USA: Oxford Univ. Press, 2000.