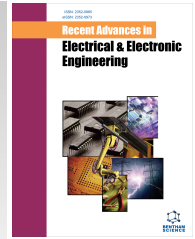


## RESEARCH ARTICLE



# An Optimized Retiming-based FIR Filter Architecture using Efficient Multipliers and Adders



Kiran Kumar Bhadavath<sup>1,\*</sup> and Z. Mary Livinsa<sup>1</sup>

<sup>1</sup>Department of ECE, Sathyabama Institute of Science and Technology, Chennai, India

**Abstract: Introduction:** An efficient higher-order filter architecture is designed and implemented for ECG signal processing applications. To prune the latency of the architecture, the retiming technique is considered for the design of less memory type Finite Impulse Response (FIR) filter architecture. The optimized multipliers and adders are key blocks in the filter architecture to increase the latency and Power Consumption (PC).

**Methods:** In this regard, an optimized Radix-4 Booth Multiplier (RBM) is designed using a modified booth encoder and selector blocks along with the proposed improved version of Square Root Carry Select Adders (SRCSLA). The design metrics of the multiplier, which is used to multiply the inputs and coefficients of the filter also improved by using the proposed SRCSLA. For the proposed SRCSLA, The Carry Look Ahead (CLA) adder and Carry Skip Adder (CSA) are combined and modified for the proposed SRCSLA. Different bit size-based SRCSLA adders are implemented for the proposed multiplier architecture and to accumulate the final filter output. The adder structure speed is improved by modified carry-producing and propagating blocks. The retiming-based FIR filter architecture with order  $N = 32$  is coded by HDL and synthesis is carried out using Genus synthesis tools in 45nm CMOS technology provided by CADENCE.

**Results:** Area Complexity (AC), Delay Complexity (DC), and PC are estimated by the reports generated by the software tool. The trade-off design metrics Area-Delay-Product (ADP) and Power-Delay-Product (PDP) are also estimated and correlated with the conventional filter structures and existing filter architectures.

**Discussion:** The proposed higher-order FIR filter architecture demonstrates significant improvements in latency and power efficiency for ECG signal processing by leveraging retiming techniques and optimized arithmetic blocks. The integration of an enhanced RBM with a novel SRCSLA, which combines features of CLA and CSA, results in faster and more power-efficient computation. Synthesized in 45nm CMOS using Cadence Genus, the design shows reduced Area, Delay, and Power. However, the complexity of SRCSLA logic and increased design overhead for retiming may present scalability and implementation challenges for ultra-high-order filters.

**Conclusion:** The proposed work is better comparatively existing works using the retiming concept.

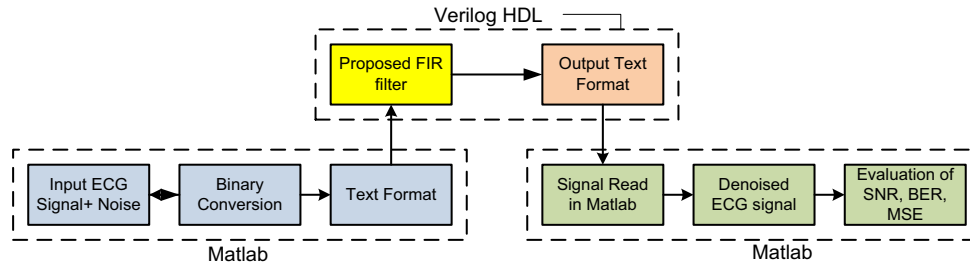
**Keywords:** Booth multiplier, CLA, CSLA, FIR filter, low power retiming, ECG signal.

## 1. INTRODUCTION

In biomedical signal processing applications, ECG signal processing is a good example for the research. ECG signal is distorted by various noise sources which causes the wrong interpretation and analysis. A denoising filter is required for the proper analysis of ECG information and

interpretation [1]. Here, a low power-area-based higher-order architecture of the FIR filter with high speed is designed for ECG denoising applications. The cost of the higher-order filter is increased by the multipliers and adders. The latency of the filter architecture is proportionally increased by the higher the filter length. The architecture of the proposed FIR filter architecture block is highlighted in denoising applications of the ECG signal, as shown in Fig. (1). Various available filter architectures for different applications are described in the literature survey of this paper.

\* Address correspondence to this author at the Department of ECE, Sathyabama Institute of Science and Technology, Chennai, India; E-mail: [kiranbadhavat@gmail.com](mailto:kiranbadhavat@gmail.com)



**Fig. (1).** Flow of the Denoising the ECG signal using FIR filter. (A higher resolution / colour version of this figure is available in the electronic copy of the article).

## 2. LITERATURE SURVEY

This section provides an overview of current filter architectures and their fundamental attributes. The Transposed Form (TF) FIR filter consists of multipliers and adders, resulting in a significantly reduced latency. With each filter tap, the length of the addition path word increases by one. Therefore, the critical path increases. As a consequence, the input capacitance and the area of the Direct Form (DF) filter both experience an increase. Hardware resource reduction may be possible with symmetric convolutions. They have not been implemented in modern transposed block FIR designs. Thus, we investigate symmetric convolution in transposed parallel FIRs and suggest a hardware-efficient parallel design. For temporal reuse of intermediate tap products, the proposed design adds delay components following multipliers. This halves the multipliers needed. We can get up to  $3.2\times$  and  $1.64\times$  area efficiency increases over the current transposed block technique for reconfigurable and fixed designs, respectively. The STB-FIR architecture for hardware-efficient, high-speed signal processing is proven by these results [2]. FIR filter computational complexity grows with length hence many methods have been devised to build practical FIR filter topologies. The multiple input multiple output (MIMO)-based parallel FIR filter architecture often requires a large area for realization as parallelism and filter order increase. The work [3] presents an architecturally enhanced, novel parallel architecture for FIR filter design that eliminates resource dependencies on parallelism. The suggested architecture limits the number of multipliers regardless of parallelism and adjusts data flow to improve iteration time. In this study, the level of parallelism ( $L$ ) is set to 8, which is neither too low nor too high. Increasing  $L$  will increase the number of samples generated at a given time. The work is shown utilizing tap 16 FIR filter on FPGA VIRTEX 4 xc4vsx35-10ff668 and VIRTEX 5 XC5VXSX95T-1FF1136 platforms and filter-order 16 on ARTIX 7 xc7a200t-2fbg676. Khoo *et al.* [4] reported that dividing the DF-FIR filter into sub-filters and limiting its cycle time can improve its efficiency. Samueli [5] used timing-driven logic optimizations to minimize filter implementation costs. Chandra and Chattopadhyay [6] reviewed several hardware-efficient FIR filter designs and found them satisfactory. The definitions of optimization algorithms were comprehensively outlined in a methodical style. The topic of sequential circuit retiming is discussed in detail [7]. Distributed arithmetic (DA)-based re-

configurable FIR filters are appropriate for SDR applications [8]. Due to the complexity of sum and carry bit creation, DA-FIR filter adder and multipliers limit area and power dissipation. PPA based on Ling equation reduces adder hardware implementation time. A field-programmable gate array implements the Verilog hardware description language (HDL)-designed MDA-RFIR filter for  $N = 64$  with 64 taps. Validated SDR channel equalization design integrates RFIR and SDR into a single system using Artix-7 development board XC7A100tCSG324. Jiang and Brayton [9] performed retiming and synthesis of logic architectures, focusing on their complexity. Yagain and Vijay [10] employed a combination of retiming and the multiplier-less filter method. Meher [11] employed cut-set and adaptable retiming techniques to create the delays of conventional DF filters, albeit resulting in increased area penalties.

Thakara *et al.* [12] constructed FIR filters using unfolding transformation, Vedic multiplication, and Carry Increment Addition. Swati and Himanshu [13] observed no significant performance improvement over DF-FIR filters in their Wallace multiplier-based FIR filter design. Krishnamurthy *et al.* [14] used a pipelined bypass multiplier with CLA adders to create an FIR filter. They achieved a latency gain over a standard DF-FIR filter with more area complexity. Rai *et al.* [15] built an FIR filter architecture for order 16 using optimized Vedic multipliers. In this work, latency and energy efficiency over the DF-FIR filter were improved with a little bit of extra hardware. Ting *et al.* [16] proposed pipelining the adaptive LMS predictor's multiplier and FIR filter with fine-grain pipelining for electronic measurement applications. The power is reduced and high throughput is achieved with more area. Meher and Park [17] increased adaptive FIR filter throughput with pipelining. They added trees to a typical FIR filter to generate two levels of pipelining, reducing critical route length while increasing area and latency. Pipelining in digital signal processing circuits has many applications, as Meher [18] has shown. Faust *et al.* [19] used transposed pipelines for structural adders in FIR filters. Pipelining delays increase hardware complexity to improve critical route length.

To accommodate the requirements of their adaptive filter design, Pramod and Shahana [20] created modified transpose form FIR filters and an improvement in CPD as well as a reduction in the adders in the CPD through the recreating

of delay taps and the successful retiming of FFs are obtained. In [21], the author discusses a quick adder that makes use of multiplexers and NOR gates. In [22], a discussion is held on low-power VLSI designs that make use of the retiming transformation approach. Recently, higher-order filter architectures have been proposed using an appropriate retiming delay in the structure [23]. Recently, one-dimensional FIR filter architectures have been proposed in [24, 25], which describes the improvement in the VLSI design metrics of the architectures using Look-Up Table (LUT) based Distributed Arithmetic (DA) logics. Next, various two-dimensional FIR filter architectures are also provided by Odugu *et al.* [26-29]. In these works, the arithmetic blocks, such as multipliers and adders, are optimized in different ways, such as Canonical Signed Digit (CSD), LUT-DA, approximate circuits, and high-speed arithmetic circuits. The modified CLA and Carry select concepts are combined to produce the high-speed adder structure to improve the design parameters of the architectures. In the existing filters, the higher throughput also increases the complexity of the filter structure. In this paper, high throughput and higher order filter architecture are proposed with less hardware complexity using retiming concept and high-speed multipliers and adders. The optimizations in the multipliers are discussed in the design of 2D FIR filters [30-36]. The redesigned, direct form register structure of FIR filters permits reuse and concurrent computing with fewer registers. The latency of DA and other conventional methods rises with filter length. This is because adders produce distinct partial products. Due to the complexity of sum and carry bit creation, adders and multipliers in DA-FIR filters limit area and power dissipation. Ling equation-based parallel prefix adder (PPA) usage reduces adder hardware implementation time. PPA uses shift-add multiplication, also known as Bypass Zero feed multiplicand in direct multiplication, to efficiently maximize area-power product. An MDA-based RFIR filter with 64 taps is designed. Field-programmable gate arrays implement Verilog hardware description language designs. Also, this design confirms SDR channel equalization.

Some of the advanced approaches for the realization of the FIR filter architectures are reported here. A narrow transition-band FIR filter is designed using back-propagation-based deep learning in this research [37-47]. The deep learning-based approach unifies FIR filter design. Analytically, the suggested technique converges when weights across neighboring layers are updated constantly. Simulations have shown the frequency response characteristics of numerous FIR filters with narrow transition bands developed using the proposed approach. Our design strategy also has magnitude response advantages over other current state-of-the-art solutions. Simulation results demonstrate improved transition bandwidth compared to a few current efforts. After being built on Altera's Cyclone IV FPGA processor, the suggested design's hardware efficiency has been proven by comparing

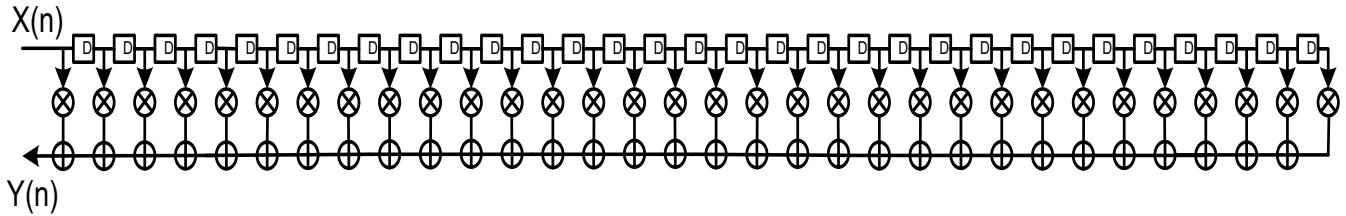
its hardware cost to various state-of-the-art FIR filters.

Complex dynamic behaviors are common in high-dimensional, high-frequency continuous-vibration control challenges. Conventional control approaches struggle to find vibration-suppressing control laws for complicated systems. A new vibration controller employing reinforcement learning (RL) and a finite-impulse-response (FIR) filter is proposed in this study [48]. The vibration mechanism was first simulated with enough physical correctness. Then, the simulator and deep deterministic policy gradient method found a near-optimal control strategy to achieve goals. Finally, real-world trials with high-dimensional and high-frequency dynamics ran the neural network control strategy directly on a controller. Traditional digital FIR filters have drawbacks. These FIR filters cannot filter undesired signals. An optimization technique for building digital FIR filters using the Back Propagation Neural Network (BPNN) algorithm is proposed in this research to overcome these constraints [49]. The linear features of digital FIR filters were used to create an amplitude response model. Then, the BPNN method minimizes the estimated error between the ideal and real amplitude response to obtain the optimal digital FIR filter coefficients. Finally, many design examples are used to test our BPNN-based optimal design. Simulations demonstrate that our BPNN-based optimal solution outperforms sequential and rectangular window designs in filtering but has higher computing complexity.

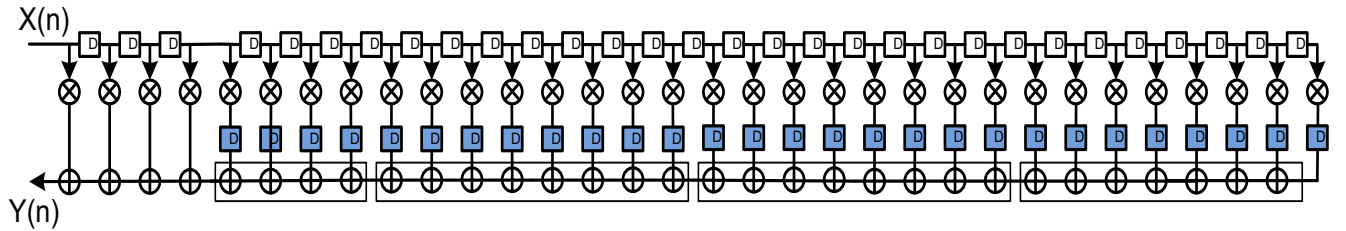
In the work [50], the design of 8 x 8 Vedic multipliers using the "UrdhvaTiryagbhyam" Sutra, "Nikhilam Sutra," and Karatsuba algorithm. It uses four four-by-four Vedic modules, an 8:1 carry-save adder, and two nine-bit binary adders to yield a 16-bit product. The UrdhvaTiryagbhyam method divides multiplications, the Nikhilam Sutra employs binary complements, and the Karatsuba technique improves huge multiplications. The suggested addition microarchitecture includes a Fast Carry Switching Adder, Kogge-Stone Adder, selection signals, and speculative logic to increase carry propagation time. An FIR filter and CNN processing element show considerable speed and efficiency improvements with the Vedic multiplier. In another work [51], a square root CLA-based filter is designed in 45nm technology to process the ECG signals. From these works, few features are also considered.

The innovative features of this paper are:

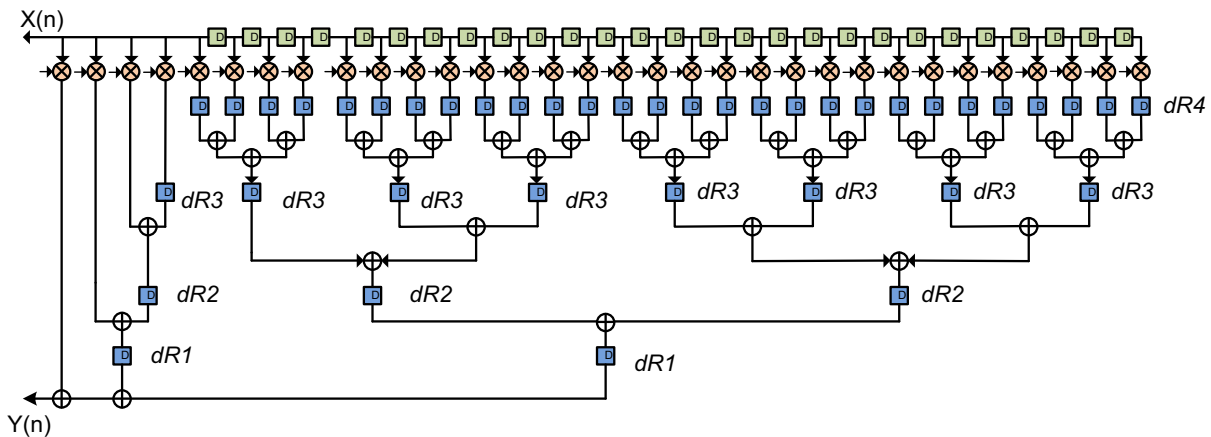
- A higher-order differential filter is utilized to construct an efficient FIR filter architecture for denoising ECG signals.
- This architecture is designed to optimize space, power, memory, and delay in ECG signal-denoising applications.
- A high-speed SRCSA is designed using redesigned carry generate and propagate modules, reducing the reliance on CPD.
- The speed of the RBM is increased by utilizing the suggested SRCSA adders for the accumulation of the partial products.



**Fig. (2).** Conventional DF architecture of FIR filter for the length  $N = 32$ . (A higher resolution / colour version of this figure is available in the electronic copy of the article).



**Fig. (3).** Normal DF architecture of FIR filter with retiming delays for the length  $N = 32$ . (A higher resolution / colour version of this figure is available in the electronic copy of the article).



**Fig. (4).** Proposed retiming-based filter architecture for  $N = 32$ . (A higher resolution / colour version of this figure is available in the electronic copy of the article).

- The concept of retiming is strategically implemented at specific locations to decrease the latency of the overall high-order filter design.

- The suggested FIR filter architecture, with a filter length of  $N = 32$ , is developed and synthesized using 45nm CMOS technology. VLSI design metrics reports are provided and compared to those of state-of-the-art filters.

**3. MATERIALS AND METHODS**

The conventional FIR filter in the direct form for  $N = 32$  is presented in Fig. (2) and the conventional filter with retiming delays is shown in Fig. (3). The architecture consists of

registers, multipliers, and adders. The overall architecture critical path is high and can be reduced by the concept of retiming. The normal Ripple Carry Adder (RCA) delay is also very high. In the proposed work, high-speed adders such as CLA and Carry Skip Adder (CSA) methodologies are combined, and a new high-speed adder SRCSLA is proposed. This proposed adder is utilized for the sum of Partial Products (PP) of the multiplier and as well as to accumulate the outputs of the individual multipliers to produce the overall output of the filter. Next, Retiming can be utilized to minimize a circuit's CPD, quantity of memory elements, and power consumption. Retiming is used in this case to shorten the CPD and hence the time of the clock without increasing

delay.

The proposed filter design for a 32-length FIR filter is shown in Fig. (4). It is realized by a 4-tap delayed production Sub-Filter (SF), a 4-tap Delayed Sub-Filter (DSF), and three eight-tap DSFs. The dR1, dR2, dR3, and dR4 represent the delays induced by retiming delays in locations 1, 2, 3, and 4, respectively. The delay dR4 separates the adders and multipliers in a DSF and restricts the delay of the multiplier, Tm. The delay dR3 limits the maximum combinational delay of DSF added to 2Ta. The FFs dR1 and dR2 guarantee that the filter's remaining section's maximum combinational route delay to the final result is equal to or less than 2Ta. It needs to be observed that for filter lengths exceeding 32, the amount of DSF taps can be raised proportionately, decreasing the taps.

The proposed filter architecture is a retimed filter structure that employs efficient SRCSLA adders and enhanced R4BM over long delay lines. The CPD of the proposed filter, taking into account the delays in positions 1, 2, 3, and 4, can be expressed as the delay corresponding to the multiplier and adder by:

$$t_f = t_{m0} + t_{af} \tag{1}$$

Where tm0, is the multiplication time to compute the coefficient-input sample product term and taf is the maximum bit-width final addition time. Equation (1) shows that utilizing a less delay multiplier and adder along the long path channels can lower CPD. Therefore, delay-efficient adders and multipliers replace the first 3-taps' adders and multipliers. Low delay adders replace all intermediate delayed DR2 and filter output adders. Partial product addition with the delay-efficient adder reduces RBM latency. RCA was replaced by a delay-efficient SRCSLA adder to enhance RBM latency. Reduced delay costs more area and power. Thus, only time-consuming routes apply the delay effectiveness multiplier.

### 3.1. Proposed Single CSLA and SRCSLA Adder Structures

A traditional CSLA is implemented using the RCA-RCA architecture, which produces carry and sum corresponding to the input carry values of Cin = 0 or 1. It then selects one of each pair to obtain the final values of carry and sum [37]. Despite having a lower CPD compared to a regular RCA, the dual RCA construction of a conventional CSLA is considered unappealing. There have been limited efforts to prohibit the use of dual RCA in CSLA design.

Instead of utilizing two RCAs, Kim and Kim [38] opted for a single RCA and MUX-based circuits. They (referred to as “et al.” in academic writing) proposed a Sqrt-CSLA as a method for effectively implementing large bit-width ad-

ders. A Sqrt-CSLA is formed by cascading CSLAs of progressively larger sizes. The primary objective of Sqrt-CSLA is to minimize the delay of adders by offering parallel carry transmission channels. Ramkumar and Kittur [40] presented a BEC-based CSLA technique. Although BEC-based CSLAs have reduced logic resource requirements, they exhibit increased latency.

An ordinary CSLA based on Boolean logic was presented [41, 42]. The CBL-based CSLA [41] utilizes fewer logic resources compared to the original CSLA while maintaining a similar CPD to the RCA. The problem is addressed by proposing CBL-based Sqrt-CSLAs [42]. This design utilizes a greater amount of logical resources and experiences longer delays compared to the BEC-based Sqrt-CSLA [43].

According to a survey of the literature, optimization in the logic relies on redundant processes corresponding to the given expression, while adder latency depends on the data. Existing designs optimize logic without considering data. This short examines traditional and BEC-based CSLA logic operations for data dependency and duplicate logic operations [46]. This paper proposes a logical CSLA formulation based on these observations. The data dependence logic formulation and a new type of Carry Generator (CG) block and Carry Select (CS) block architecture in Fig. (5) are the briefs main contributions.

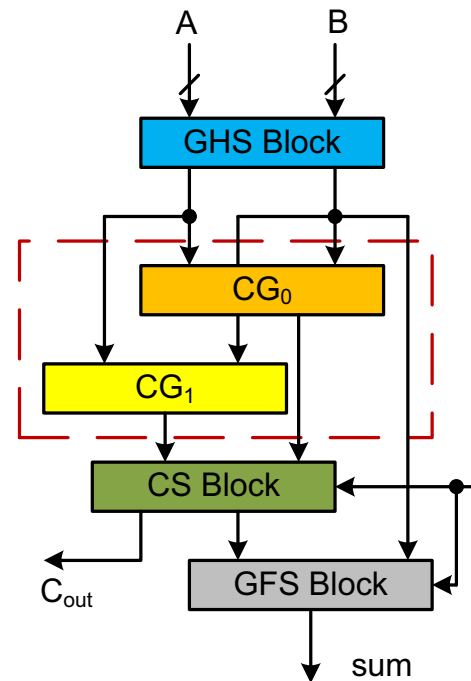
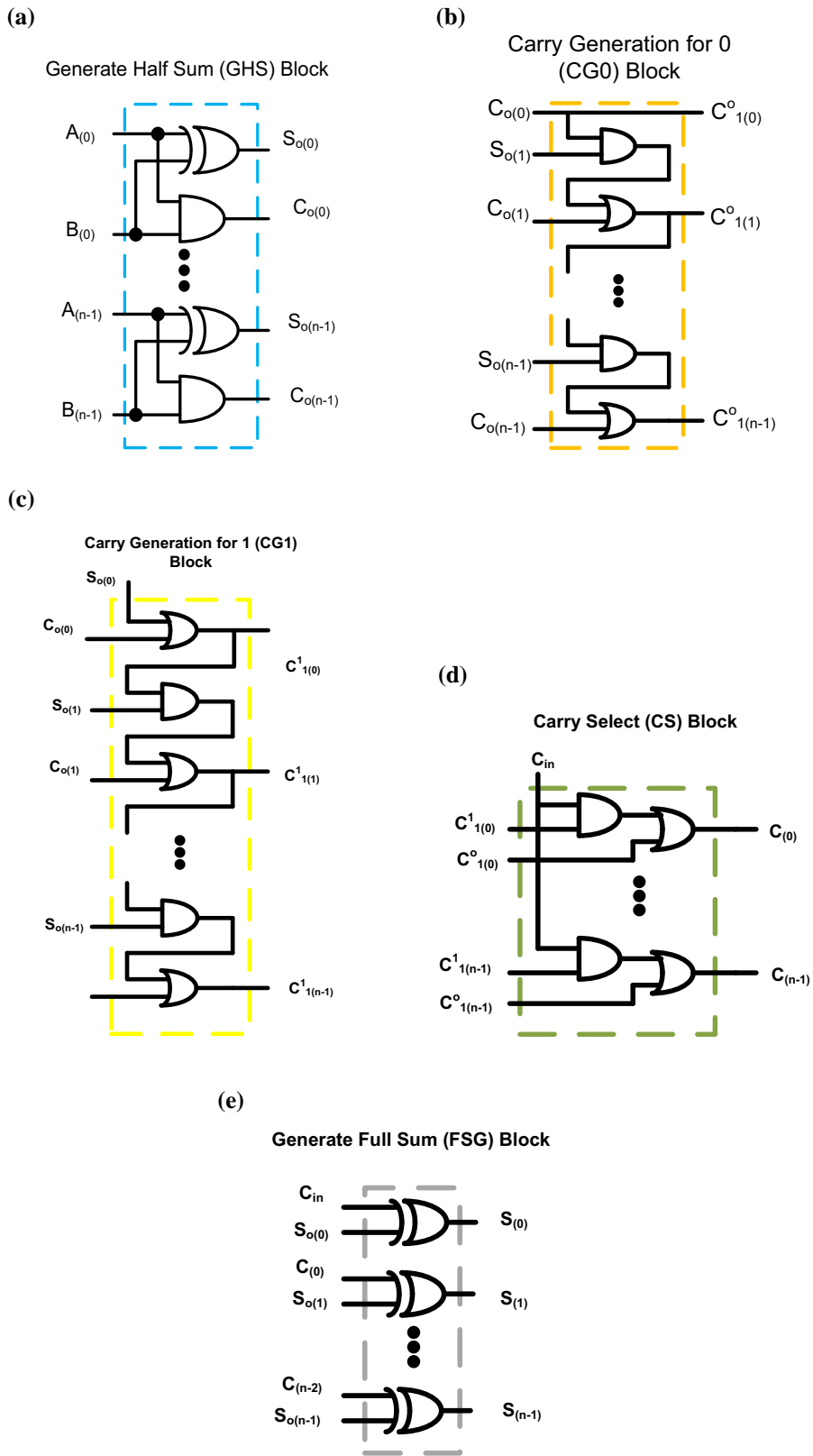


Fig. (5). Proposed CSLA adder for adding the individual multiplier outputs. (A higher resolution / colour version of this figure is available in the electronic copy of the article).



**Fig. (6).** Individual blocks of CSLA. (a) Generate half sum (GHS) block; (b) Carry generation for 0 (CG0) block; (c) Carry generation for 1 (CG1) block; (d) Carry select (CS) block; (e) Generate full sum (FSG) block. (A higher resolution / colour version of this figure is available in the electronic copy of the article).

The logical expressions for the proposed CSLA are expressed in equations (2-9),

$$S_0(i) = A(i) \oplus B(i) \quad (2)$$

$$C_0(i) = A(i).B(i) \quad (3)$$

$$C_1^0(i) = C_1^0(i-1).S_0(i) + C_0(i) \quad \text{for } [C_1^0(0) = 0] \quad (4)$$

$$C_1^1(i) = C_1^1(i-1).S_0(i) + C_0(i) \quad \text{for } [C_1^1(0) = 1] \quad (5)$$

$$C(i) = C_1^0(i) \quad \text{if } (C_{in} = 0) \quad (6)$$

$$C(i) = C_1^1(i) \quad \text{if } (C_{in} = 1) \quad (7)$$

$$C_{out} = C(n-1) \quad (8)$$

$$S(0) = S_0(0) \oplus C_{in} \quad S(i) = S_0(i) \oplus C(i-1) \quad (9)$$

Fig. (5) depicts the structure that corresponds to the expressions mentioned above. One Generate Full Sum (GFS) block, one Generate Half Sum (GHS) block, one CG block, and one CS block comprise it. Two CG blocks, such as CG0 block and CG1 block, which correspond to the input carry of '0' and '1' comprise the CG block. Given two n-bit operands (A and B), the GHS gives a half-carry C0 and a half-sum S0, both of which have a width of n bits. Upon receiving S0 and C0 from the GHS, CG0, and CG1 produce two n-bit words of the full Carry, denoted as C10 and C11, which are equivalent to the input-carry values '0' and '1' correspondingly. In Fig. (6a), the structural diagram of the GHS block is illustrated.

The CG0 and CG1 logic blocks are tuned to make use of the fixed input-carry bits. Fig. (6b and c) illustrate the mixed logic designs of CG0 and CG1, respectively. According to the Cin signal, the CS block picks one of the carry words as the final carry from the available two carry words on its input line. When Cin = 0, it chooses C10; otherwise, it chooses C11. An n-bit 2x1 MUX can be used to implement

the CS unit. However, we can see from the CS block's truth table that carry-words C10 and C11 follow a precise bit pattern. If C10i=1, then C11i=1, regardless of S0(i) or C0(i), for 0 ≤ i ≤ n-1.

This functionality is used to optimize the logic of the CS block. Fig. (6d) shows the best CS block design of n AND-OR gates. CS contains the last carry word 'C'. Fig. (6e) shows how the FSG XORs (n-1) LSBs with (n-1) MSBs of half-sum (S0) to form (n-1) MSBs of final-sum (S). S0's LSB is XORed with Cin to get S's.

SRCSLA, a chain of CSLAs, completely leverages multi-path carry propagation. To maximize carry propagation channel concurrence, the SRCSLA uses CSLAs of increasing size. The SRCSLA architecture constructs huge adders with significantly reduced latency compared to a single-stage CSLA. The carry propagation time between the SRCSLA and CSLA stages has an impact on the total latency of the adder. The recommended CSLA architecture is preferable to current designs for area-delay efficient SRCSLA implementation because it incorporates early output carry with multi-path carry propagation. Fig. (7) demonstrates a 16-bit SRCSLA design, which incorporates the 2-bit RCA, 2-bit CSLA, 3-bit CSLA, 4-bit CSLA, and 5-bit CSLA from the planned CSLA. To ascertain the filter output, the outputs of the multipliers are summed utilizing this 16-bit SRCSLA.

### 3.2. Proposed Structure of the RBM

Booth multipliers have been increasingly popular in recent years because of their effective use of space, little delay, and energy-saving design. The primary concept behind Booth multiplication is to utilize the Booth encoding technique to reduce the number of incomplete products. The conventional Booth multiplier architecture exhibits increased latency due to the sequential accumulation of partial products and occupies a larger physical footprint as a result of the involvement of numerous logic resources. Consequently, numerous endeavors have been undertaken in academic

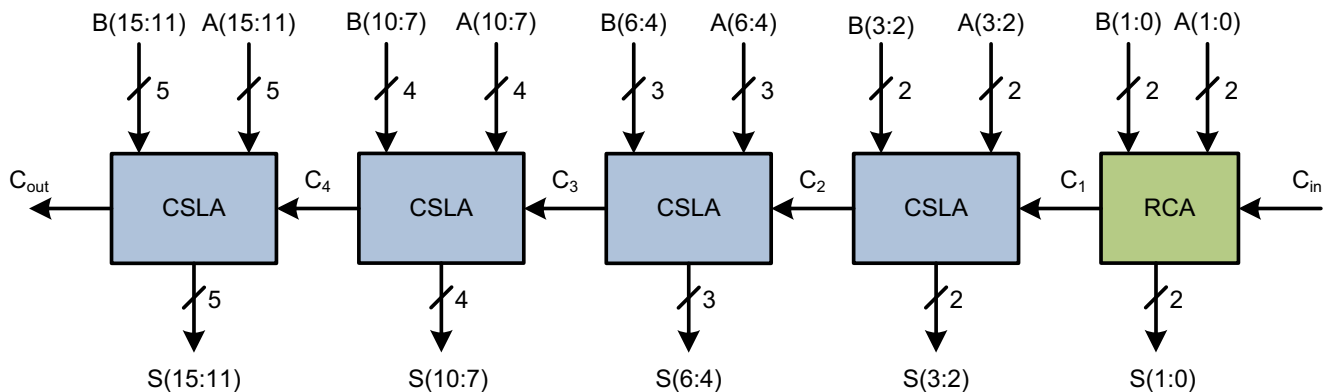
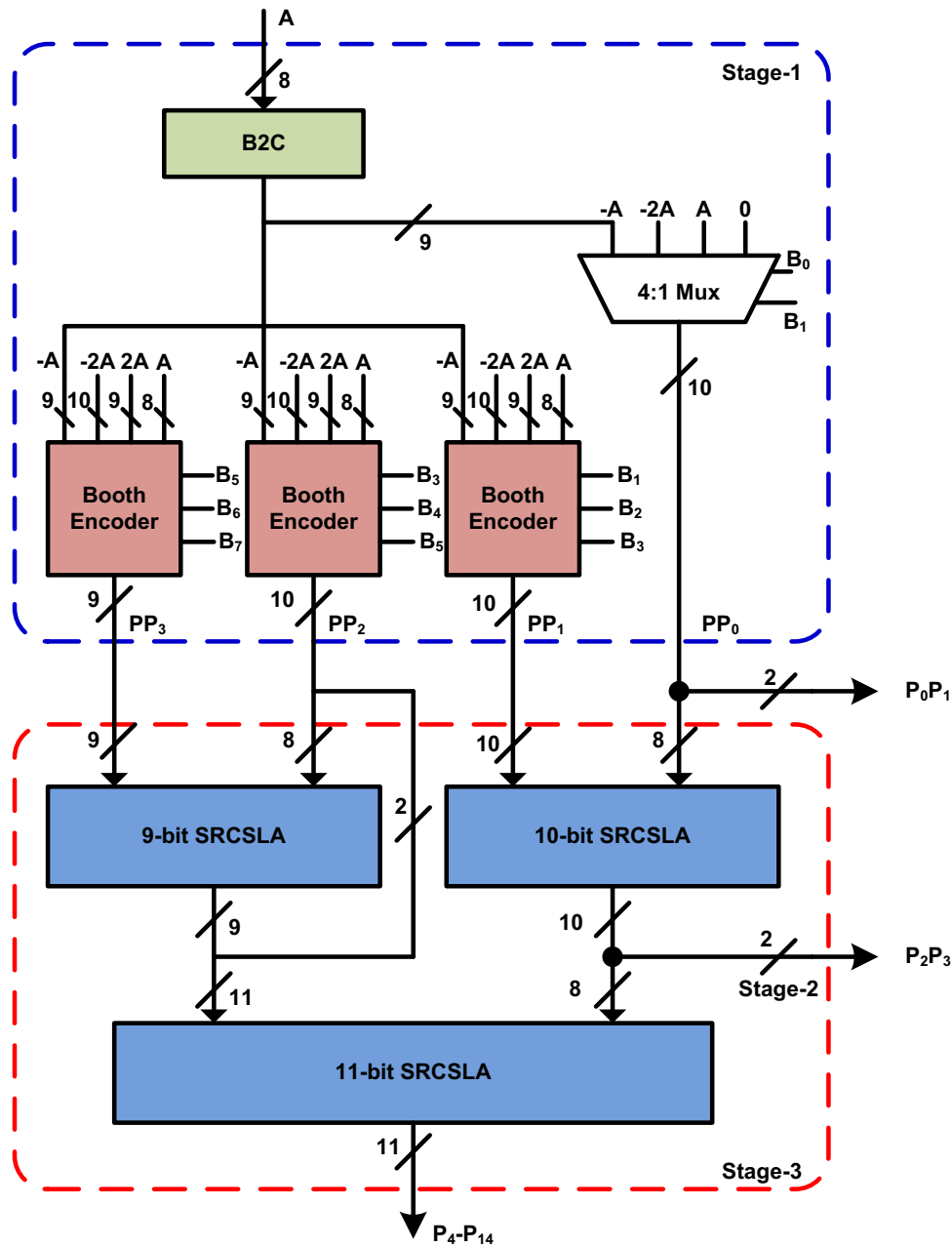


Fig. (7). Square Root CSLA (SRCSLA) for n = 16. (A higher resolution / colour version of this figure is available in the electronic copy of the article).



**Fig. (8).** Architecture of the proposed 8×8 RBM using the proposed SRCSLA. (A higher resolution / colour version of this figure is available in the electronic copy of the article).

literature to develop Booth multipliers that are effective in terms of both area and latency. The following are the significant advancements in this field over the previous two decades. The R4BM designs were recently presented [44, 45]. Two designs for area and delay optimization are proposed [45]. However, for the addition of encoded partial products, these systems require 15-bit adders, which doubles the complexity. The 8×8 RBM performs the simultaneous addition of encoded partial products to reduce time. They updated the Booth Encoder (BE) and Binary-to-2's Complement Converter (B2CC) designs for this arrangement.

In addition, the RBM design used a new CLA-based CS-LA. However, in a study [45], BM design has redundant logic. Thus, the current Booth multiplier structure [45] needs latency and area improvements. The next section examines this study's [45] structure and several internal components to find the duplicate logic. This section thoroughly analyzes [45-50]'s latest BM design for area-delay minimization (Fig. 8). The system includes B2CC, three BEs, a 4x1 MUX, and Adder Trees (AT). This design multiplies A and B in three steps. The first phase creates radix-4 Booth-encoded PPs, which are added in the 2<sup>nd</sup> and 3<sup>rd</sup> steps. Thus, Figs. (9-11)

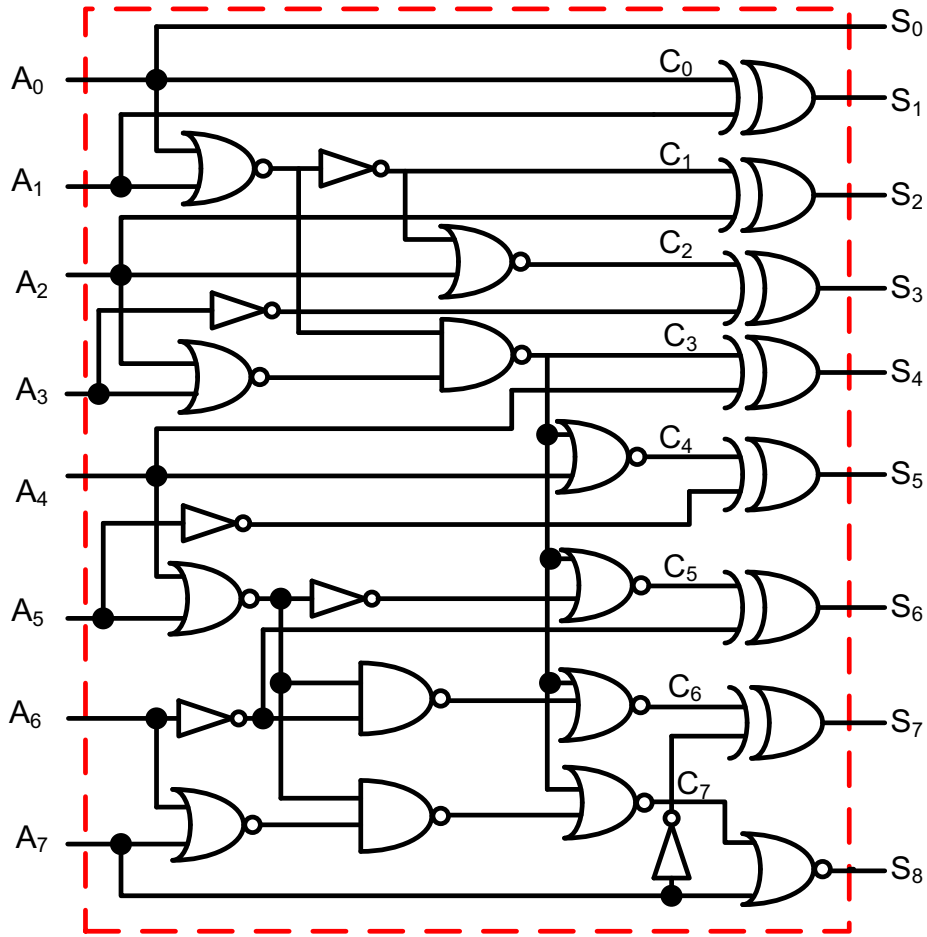


Fig. (9). Internal structure of the binary to 2's complement converter (B2CC) block. (A higher resolution / colour version of this figure is available in the electronic copy of the article).

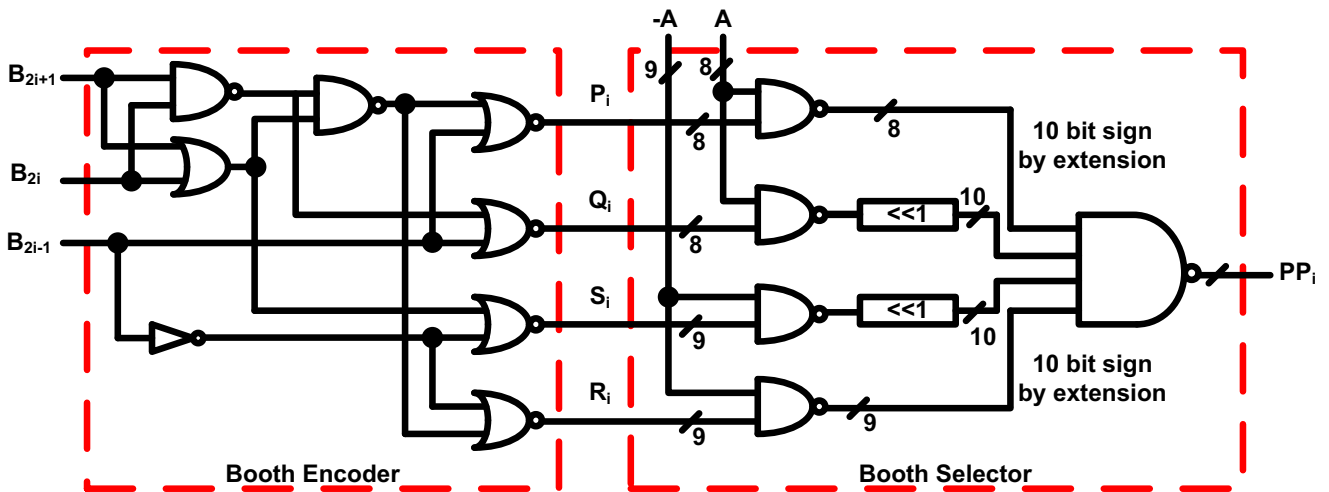
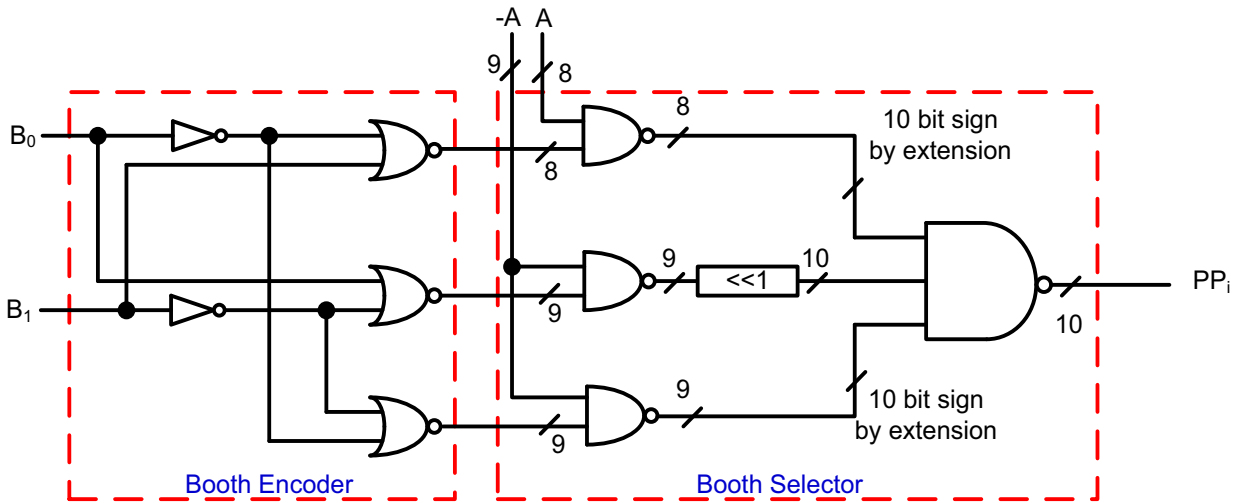
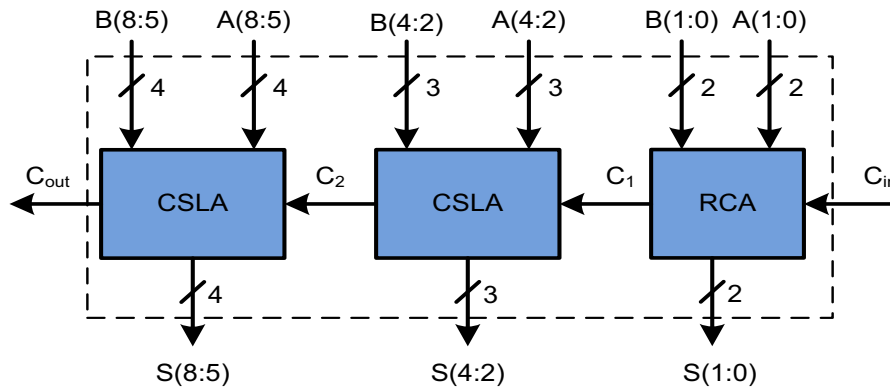


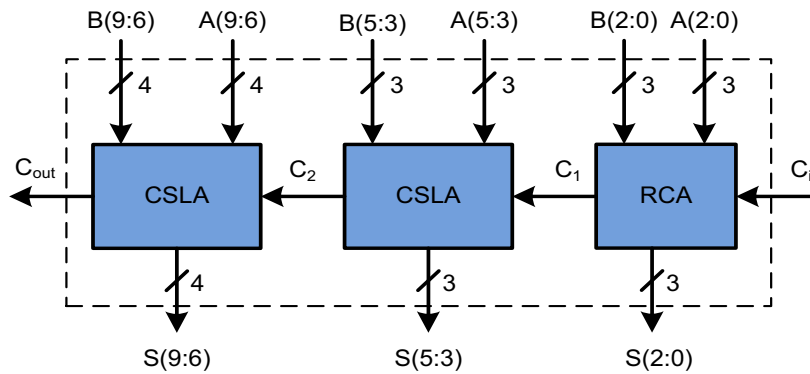
Fig. (10). Gate level structure of the booth encoder (BE) and selector for the generation of partial product of  $PP_i$ . (A higher resolution / colour version of this figure is available in the electronic copy of the article).



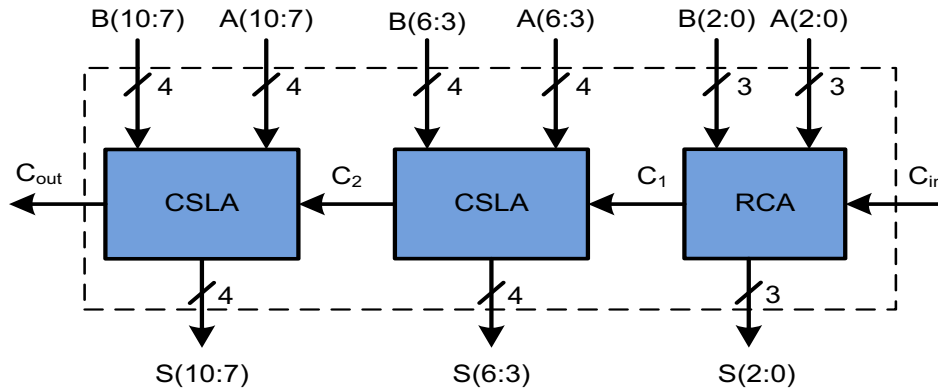
**Fig. (11).** Gate level structure of the BE and selector for the generation of partial product of PP0. (A higher resolution / colour version of this figure is available in the electronic copy of the article).



**Fig. (12).** Architecture of the 9-bit SRCSLA for Booth multiplier. (A higher resolution / colour version of this figure is available in the electronic copy of the article).



**Fig. (13).** Architecture of the 10-bit SRCSLA for Booth multiplier. (A higher resolution / colour version of this figure is available in the electronic copy of the article).



**Fig. (14).** Architecture of the 11-bit SRCSLA for Booth multiplier. (A higher resolution / colour version of this figure is available in the electronic copy of the article).

**Table 1.** Performance parameters comparison of the filter architecture with conventional filters.

Architecture Type	N	Power (mW)	Area ( $\mu\text{m}^2$ )	Delay (ns)	ADP ( $\mu\text{m}^2 \cdot \mu\text{s}$ )	PDP (mW. ns)
Conventional FIR Filter Architecture	32	1.5584	38357.21	1.8597	71.332	2.898
Normal FIR Filter architecture with Retiming	32	0.9839	33968.62	1.5402	52.318	1.515
Proposed FIR Filter Architecture	32	0.5112	22236.41	0.9868	21.942	0.504

show each unit's internal structures for Booth multiplier design optimization. The Booth multiplier requires 9-bit, 10-bit, and 11-bit SRCSLA adders, as shown in Figs. (12-14)

#### 4. RESULTS AND DISCUSSION

The proposed retiming-based higher order FIR filter architecture using optimized booth multipliers and high-speed SRCSLA adders are coded using structural Verilog HDL and synthesized by Cadence-Genus Synthesis tool in 45 nm CMOS technology library. The reports related to the proposed VLSI architecture are generated and tabulated for the proposed DF-FIR filter for  $N = 32$ . The conventional FIR filter architecture without any retiming registers with normal multipliers and RCA adders was also implemented for comparison purposes. Subsequently, a different FIR filter structure is constructed using standard adders and multipliers, and it is synchronized with the proposed FIR filter design by proper retiming. Table 1 presents a comparison of these three types of filters based on their area, timing delay, power, ADP, and PDP.

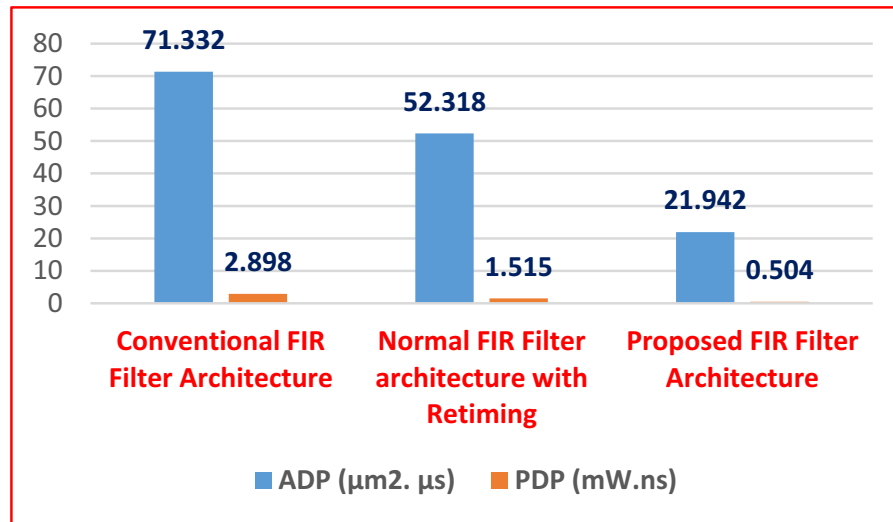
The retiming-based filter architecture and upgraded multipliers and adders generated efficient area, delay, power, ADP, and PDP values compared to conventional FIR filter construction and regular filter architecture with retiming. Fig. (15) shows the ADP and PDP values of the suggested filter implementation with retiming and optimized adders and multipliers compared to a conventional and standard filter with retiming. The suggested filter architecture reduces PDP by 82.6% and 66.73% and ADP by 69.23% and

58.06% compared to conventional and retiming-based normal filters. The ADP and PDP values are reduced due to the innovative modifications in the RBM using new encoder and selector blocks along with optimized SRCSLAs.

The performance characteristics of the proposed retiming-based power and delay-efficient FIR filter realization using high-speed adders & multipliers are compared to existing filter architectures with  $N = 32$  in Table 2.

The suggested FIR filter architecture reduces ADP by 85% to 25% compared to Chulet *et al.* [13] and Patali *et al.* [23] filter-2. The suggested filter architecture reduces PDP by 92% compared to Chulet *et al.* [13] and 31% compared to Patali *et al.*'s [23]. filter-2. These metrics are improved by the innovative designs used for the implementation of arithmetic blocks of the filter and retiming in the filter architecture.

**Advantages:** The proposed filter architecture delay is very less comparatively other filter architectures due to the retiming of the FFs. The critical path delay is reduced in the multiplier also using high speed adders. The architecture of the filter is compatible with any application for filter processing. The redundant blocks are reduced in the multipliers and adders designed to reduce the area and power of the architecture. These design styles are computable for higher filter lengths also. When compared to the learning network-based FIR filter architectures, the proposed filter architecture power consumption, area, and delay are much less. In the learning networks, more layers consist of more adders, and hence, the area and power parameters are high.



**Fig. (15).** The ADP and PDP of the suggested and conventional filter architectures comparison. (A higher resolution / colour version of this figure is available in the electronic copy of the article).

**Table 2.** Compares the proposed filter architecture to N = 32 current filter architectures.

Filter Architectures	(N)	Power (mW)	Area ( $\mu\text{m}^2$ )	Delay (ns)	ADP ( $\mu\text{m}^2 \cdot \mu\text{s}$ )	PDP (mW. ns)
Meher [11] filter-1	32	0.658	29207	4.53	132.307	2.980
Meher [11] filter-2	32	0.714	28379	2.48	70.379	1.770
Chulet <i>et al.</i> [13]	32	1.385	30786	4.87	149.927	6.744
Rai <i>et al.</i> [15]	32	1.144	35953	1.97	70.827	2.253
Meher and Park [17]	32	0.562	23665	1.22	28.871	0.685
Pramod <i>et al.</i> [20]	32	0.514	24661	1.47	36.251	0.755
Patali <i>et al.</i> [23] filter-1	32	0.622	26181	1.28	33.511	0.796
Patali <i>et al.</i> [23] filter-2	32	0.662	26626	1.11	29.554	0.734
Sridevi & Sathiya [50]		248.93	-	27.95	-	6957.5
Ye <i>et al.</i> [2]	64	63.45	979,802	0.00483	4732.4	0.306
Proposed FIR Filter Architecture	32	0.5112	22236	0.9868	21.942	0.504

## CONCLUSION

The power-delay-efficient higher-order FIR filter architecture is presented using modified R4BMs and improved SRCSLA, along with the retiming concept for the denoising applications. The CPD of the FIR filter architecture is reduced by the retiming at the different stages of the filter structure. The traditional adders are substituted with the proposed high-speed SRCSLA adders. The BM delay and PC are decreased by the different sizes of the proposed SRCSLA adders and modified logic modules. The DF-FIR filter architecture is implemented with a length of 32 and input and coefficient sizes are considered as the 8-bits. The  $8 \times 8$  Radix-4 Booth Multiplier structures and 9, 10, 11, and 16-bit SRCSLA adders are designed for the addition of the sum of the PPs of the multiplier and to produce the final filter output, respectively. The designed FIR filter architecture is simulated and synthesized using Cadence 45nm CMOS technology-based ASIC tools. The area, delay, and power reports are analyzed and ADP and PDP metrics are evaluated and

compared with the existing and conventional filter architectures. The ADP of our filter architecture values are pruned by 85% (max) and 25% (min), and PDP values are decreased by 92% (max) and 31% (min) when compared to the existing works of the same filter length  $N = 32$ . This proposed filter architecture is well-suitable for ECG signal-denoising applications.

## STUDY LIMITATIONS

- The integration of SRCSLA increases design complexity and control logic overhead.
- Retiming adds extra design effort and may complicate timing closure for very high-order filters.
- The architecture may not scale efficiently for filters with orders significantly higher than  $N = 32$ .
- Power savings might diminish under low activity or sparse coefficient conditions.

•Area benefits may reduce when targeting larger bit-widths or multi-channel processing scenarios.

#### AUTHORS' CONTRIBUTIONS

It is hereby acknowledged that all authors have accepted responsibility for the manuscript's content and consented to its submission. They have meticulously reviewed all results and unanimously approved the final version of the manuscript.

#### LIST OF ABBREVIATIONS

AT	=	Adder Trees
AC	=	Area Complexity
ADP	=	Area-Delay-Product
BPNN	=	Back Propagation Neural Network
BE	=	Booth Encoder
CSD	=	Canonical Signed Digit
CG	=	Carry Generator
CLA	=	Carry Look Ahead
CS	=	Carry Select
CSA	=	Carry Skip Adder
DC	=	Delay Complexity
DSF	=	Delayed Sub-Filter
DF	=	Direct Form
FIR	=	Finite Impulse Response
GFS	=	Generate Full Sum
GHS	=	Generate Half Sum
LUT	=	Look-Up Table
RCA	=	Ripple Carry Adder
TF	=	Transposed Form

#### CONSENT FOR PUBLICATION

Not applicable.

#### AVAILABILITY OF DATA AND MATERIALS

The data and supportive information are available within the article.

#### FUNDING

None

#### CONFLICT OF INTEREST

The authors declare no conflict of interest, financial or otherwise.

#### ACKNOWLEDGEMENTS

Declared none.

#### REFERENCES

- [1] T.V. Padmavathy, S. Saravanan, and M.N. Vimalkumar, "Partial product addition in Vedic design-ripple carry adder design fir filter architecture for electro cardiogram (ECG) signal de-noising application", *Microprocess. Microsyst.*, vol. 76, p. 103113, 2020. <http://dx.doi.org/10.1016/j.micpro.2020.103113>
- [2] J. Ye, M. Yanagisawa, and Y. Shi, "Scalable hardware efficient architecture for parallel fir filters with symmetric coefficients", *Electronics*, vol. 11, no. 20, p. 3272, 2022. <http://dx.doi.org/10.3390/electronics11203272>
- [3] K.D. Shinde, "Optimizing Parallel FIR Filter Architecture for Time-Sensitive Applications: A Design Approach for High-Throughput and Area Efficiency", *Int. J. Intell. Eng. Syst.*, vol. 16, no. 4, 2023.
- [4] K.Y. Khoo, Z. Yu, and N.W. Alan, "Design of optimal hybrid form FIR filter", *The 2001 IEEE International Symposium on Circuits and Systems*. Sydney, NSW, Australia, 06-09 May 2001, pp. 621-624 <http://dx.doi.org/10.1109/ISCAS.2001.921147>
- [5] H. Samueli, "An improved search algorithm for the design of multiplierless FIR filters with powers-of-two coefficients", *IEEE Trans. Circuit. Syst.*, vol. 36, no. 7, p. 1044, 1989. <http://dx.doi.org/10.1109/31.31347>
- [6] A. Chandra, and S. Chattopadhyay, "Design of hardware efficient FIR filter: A review of the state-of-the-art approaches", *Int. J. Eng. Sci. Technol.*, vol. 19, no. 1, pp. 212-226, 2016. <http://dx.doi.org/10.1016/j.jestch.2015.06.006>
- [7] C.E. Leiserson, and J.B. Saxe, "Retiming synchronous circuitry", *Algorithmica*, vol. 6, no. 1-6, pp. 5-35, 1991. <http://dx.doi.org/10.1007/BF01759032>
- [8] B.N.M. Kumar, and H.G. Rangaraju, "Design and implementation of pervasive DA based FIR filter and feeder register based multiplier for software definedradio networks", *Int. J. Pervas. Comput. Commun.*, vol. 18, no. 1, pp. 43-58, 2022. <http://dx.doi.org/10.1108/IJPCC-04-2021-0086>
- [9] J.H.R. Jiang, and R.K. Brayton, "Retiming and resynthesis: A complexity perspective", *IEEE Trans. Comput. Aided Des. Integrated Circ. Syst.*, vol. 25, no. 12, pp. 2674-2686, 2006. <http://dx.doi.org/10.1109/TCAD.2006.882520>
- [10] D. Yagain, and A. Vijaya Krishna, "Design of synthesizable, retimed digital filters using FPGA-based path solvers with MCM approach: Comparison and CAD tool", *VLSI Des.*, vol. 2014, no. 1, pp. 1-18, 2014. <http://dx.doi.org/10.1155/2014/280701>
- [11] P.K. Meher, "On efficient retiming of fixed-point circuits", *IEEE Trans. Comput. Aided Des. Integrated Circ. Syst.*, vol. 24, no. 4, pp. 1257-1265, 2015. <http://dx.doi.org/10.1109/TVLSI.2015.2453324>
- [12] S. Thakral, D. Goswami, R. Sharma, C.K. Prasanna, and A.M. Joshi, "Design and implementation of a high-speed digital FIR filter using unfolding", *IEEE 7th Power India International Conference (PIICON)* Bikaner, India, 25-27 November 2016, pp. 1-4. <http://dx.doi.org/10.1109/POWERI.2016.8077361>
- [13] S. Chulet, and H. Joshi, "FIR filter designing using Wallace multiplier", *Int. J. Eng. Tech. Res.*, vol. 3, no. 6, 2015.
- [14] S. Krishnamurthy, R. Kannan, E.A. Yahya, and K. Bingi, "Design of FIR filter using novel pipelined bypass multiplier", *3rd International Symposium in Robotics and Manufacturing Automation (ROMA)* Kuala Lumpur, Malaysia, 2017, pp. 1-6. <http://dx.doi.org/10.1109/ROMA.2017.8231838>
- [15] N.S. Rai, P. Shree, Y.P. Meghana, A.P. Chavan, and H.R. Aradhya, "Design and implementation of 16-tap FIR filter for DSP applications", *2018 Second International Conference on Advances in Electronics, Computers and Communications (ICAEECC)* Bangalore, India, 09-10 February 2018, pp. 1-5. <http://dx.doi.org/10.1109/ICAEECC.2018.8479480>
- [16] L.K. Ting, R. Woods, and C.F. Cowan, "Virtex FPGA implementation of a pipelined adaptive LMS predictor for electronic support measures receivers", *IEEE T VLSI SYST*, vol. 13, no. 1, pp. 86-95, 2005.
- [17] P.K. Meher, and S.Y. Park, "Critical-path analysis and low-complexity implementation of the LMS adaptive algorithm", *IEEE*

- Trans. Circuits Syst. I Regul. Pap.*, vol. 61, no. 3, pp. 778-788, 2014.  
<http://dx.doi.org/10.1109/TCSI.2013.2284173>
- [18] P.K. Meher, "Seamless pipelining of DSP circuits", *Circuits Syst. Signal Process.*, vol. 35, no. 4, pp. 1147-1162, 2016.  
<http://dx.doi.org/10.1007/s00034-015-0089-2>
- [19] M. Faust, M. Kumm, C.H. Chang, and P. Zipf, "Efficient structural adder pipelining in transposed form FIR filters", *2015 IEEE International Conference on Digital Signal Processing (DSP)* Singapore, 2015, pp. 311-314.  
<http://dx.doi.org/10.1109/ICDSP.2015.7251882>
- [20] P. Pramod, and T.K. Shahana, "High throughput adaptive filter architecture using modified transpose form FIR filters", *J. Adv. Res. Dyn. Control Syst.*, vol. 10, no. 15, pp. 68-82, 2018.  
<http://dx.doi.org/10.1049/iet-cds.2019.0130>
- [21] H. Tavakolaee, G. Ardeshtir, and Y. Baleghi, "Fast mux-based adder with low delay and low PDP", *J. Artif. Intell. Data Min.*, vol. 7, no. 3, pp. 385-392, 2019.
- [22] S. Jalaja, "Different retiming transformation technique to design optimized low-power VLSI architecture", *AIMS Electron. Elect. Eng.*, vol. 2, no. 4, p. 117, 2018.  
<http://dx.doi.org/10.3934/ElectrEng.2018.4.117>
- [23] P. Patali, and S. Thottathikkulam Kassim, "High throughput FIR filter architectures using retiming and modified CSLA based adders", *IET Circuits Dev. Syst.*, vol. 13, no. 7, pp. 1007-1017, 2019.  
<http://dx.doi.org/10.1049/iet-cds.2019.0130>
- [24] K. Kumar Bhadavath, and Z.M. Livinsa, "An efficient approximation look up table based distributed arithmetic (DA) VLSI architecture for finite impulse response", *2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)* Greater Noida, India, 2022, pp. 1849-1852.  
<http://dx.doi.org/10.1109/ICACITE53722.2022.9823908>
- [25] K.K. Bhadavath, and Z.M. Livinsa, "Very large scale integration implementation of efficient finite impulse response filter architectures using novel distributed arithmetic for digital channelizer of software defined radio", *Int. J. Circuit Theory Appl.*, vol. 51, no. 3, pp. 1153-1167, 2023.  
<http://dx.doi.org/10.1002/cta.3467>
- [26] V.K. Odugu, V.N. C, and S.P. K, "An efficient VLSI architecture of 2D finite impulse response filter using enhanced approximate compressor circuits", *Int. J. Circuit Theory Appl.*, vol. 49, no. 11, pp. 3653-3668, 2021.  
<http://dx.doi.org/10.1002/cta.3114>
- [27] V.K. Odugu, C.V. Narasimhulu, and K.S. Prasad, "A novel filter-bank architecture of 2D-FIR symmetry filters using LUT based multipliers", *Integration (Amst.)*, vol. 84, pp. 12-25, 2022.  
<http://dx.doi.org/10.1016/j.vlsi.2022.01.004>
- [28] V.K. Odugu, C. Venkata Narasimhulu, and K. Satya Prasad, "Design and implementation of low complexity circularly symmetric 2D FIR filter architectures", *Multidimens. Syst. Signal Process.*, vol. 31, pp. 1385-1410, 2020.  
<http://dx.doi.org/10.1007/s11045-020-00714-3>
- [29] V.K. Odugu, C.V. Narasimhulu, and K.S. Prasad, "Implementation of low power and memory-efficient 2D FIR filter architecture", *Int. J. Recent. Technol. Eng.*, vol. 8, no. 1, pp. 927-935, 2019.
- [30] V.K. Odugu, C.V. Narasimhulu, and K.S. Prasad, "Implementation of low power generic 2D FIR filter bank architecture using memory-based multipliers", *J. Mobile Multimedia*, vol. 18, no. 3, pp. 583-602, 2022.  
<http://dx.doi.org/10.13052/jmm1550-4646.1836>
- [31] V.K. Odugu, J.R. B, H.B. Gade, and A.R. Uppala, "Higher order and optimized symmetric 2D FIR filter design and hardware architecture implementation using CSD-CSE", *Comput. Electr. Eng.*, vol. 119, p. 109564, 2024.  
<http://dx.doi.org/10.1016/j.compeleceng.2024.109564>
- [32] V.K. Odugu, H.B. Gade, and B. Janardhana Rao, "Efficient VLSI Architectures of Multimode 2D FIR Filter Bank using Distributed Arithmetic Methodology", In: *International Conference on Soft Computing and Signal Processing*, Springer, pp. 243-253.  
[http://dx.doi.org/10.1007/978-981-99-8451-0\\_21](http://dx.doi.org/10.1007/978-981-99-8451-0_21)
- [33] V.K. Odugu, B.J. Rao, S. Bojjawar, and U.A. Raju, "Design and implementation of optimized 2D FIR symmetric filter architecture using modified McClellan transformation and CSD-CSE", *Multidimens. Syst. Signal Process.*, vol. 36, no. 1, pp. 1-24, 2025.  
<http://dx.doi.org/10.1007/s11045-024-00895-1>
- [34] S.S. Sattiraju, and V.K. Odugu, "FPGA-based finite impulse response filter architecture design with distributed arithmetic logic to reduce power and area", *IEEE Region 10 Symposium (TEN-SYMP)* New Delhi, India, 27-29 September 2024, pp. 1-5,  
<http://dx.doi.org/10.1109/TENSYMP61132.2024.10751818>
- [35] V.S. Reddy, A.V. Juliet, E.R. Thuraka, and V.K. Odugu, "Implementation of block-based diagonal and quadrantal symmetry type 2D-FIR filter architectures using DA technique", *Comput. Electr. Eng.*, vol. 118, p. 109301, 2024.  
<http://dx.doi.org/10.1016/j.compeleceng.2024.109301>
- [36] V. Srilatha Reddy, A. Vimala Juliet, E.R. Thuraka, and V.K. Odugu, "Design and implementation of power and area efficient architectures of circular symmetry 2-D FIR filters using CSOA-based CSD", *Multidimens. Syst. Signal Process.*, vol. 35, no. 1, pp. 37-63, 2024.  
<http://dx.doi.org/10.1007/s11045-024-00887-1>
- [37] C.S. Murthy, and K. Sridevi, "Optimized DA-reconfigurable FIR filters for software defined radio channelizer applications", *Circuit World*, vol. 47, no. 3, pp. 252-261, 2021.  
<http://dx.doi.org/10.1108/CW-11-2020-0332>
- [38] Y. Kim, and L.S. Kim, "64-bit carry-select adder with reduced area", *Electron. Lett.*, vol. 37, no. 10, pp. 614-615, 2001.  
<http://dx.doi.org/10.1049/el:20010430>
- [39] Y. He, C.H. Chang, and J. Gu, "An area-efficient 64-bit square root carry-select adder for low-power applications", *2005 IEEE International Symposium on Circuits and Systems (ISCAS)* Kobe, Japan, 23-26 May 2005, pp. 4082-4085  
<http://dx.doi.org/10.1109/ISCAS.2005.1465528>
- [40] B. Ramkumar, and H.M. Kittur, "Low-power and area-efficient carry select adder.", *IEEE Trans. Large. Scale. Integ. (VLSI) Syst.*, vol. 20, no. 2, pp. 371-375, 2011.  
<http://dx.doi.org/10.1109/TVLSI.2010.2101621>
- [41] I.C. Wey, C.C. Ho, Y.S. Lin, and C.C. Peng, "An area-efficient carry select adder design by sharing the common Boolean logic term", *Proc. Imecs*, vol. 10, pp. 1-4, 2012.
- [42] S. Manju, and V. Sornagopal, "An efficient SQRT architecture of carry select adder design by common Boolean logic", *2013 International Conference on Emerging Trends in VLSI, Embedded System, Nano Electronics and Telecommunication System (ICEVENT)* Tiruvannamalai, India, 07-09 January 2013, pp. 1-5  
<http://dx.doi.org/10.1109/ICEVENT.2013.6496590>
- [43] B.K. Mohanty, and S.K. Patel, "Efficient very large scale integration architecture for variable length block least mean square adaptive filter", *IET Signal Process.*, vol. 9, no. 8, pp. 605-610, 2015.  
<http://dx.doi.org/10.1049/iet-spr.2014.0424>
- [44] P. Patali, and S. Thottathikkulam Kassim, "Efficient modular hybrid adders and Radix-4 booth multipliers for DSP applications", *Microelectronics J.*, vol. 96, p. 104701, 2020.  
<http://dx.doi.org/10.1016/j.mejo.2020.104701>
- [45] N.V.V.K. Boppana, J. Kommareddy, and S. Ren, "Low-cost and high-performance 8x 8 booth multiplier", *Circuits Syst. Signal Process.*, vol. 38, no. 9, pp. 4357-4368, 2019.  
<http://dx.doi.org/10.1007/s00034-019-01044-x>
- [46] B.K. Mohanty, and S.K. Patel, "Area-delay-power efficient carry-select adder", *IEEE Trans. Circuits Syst. II Express Briefs*, vol. 61, no. 6, pp. 418-422, 2014.  
<http://dx.doi.org/10.1109/TCSII.2014.2319695>
- [47] S. Roy, and A. Chandra, "A deep learning approach for the design of narrow transition-band FIR filter", *Circuits Syst. Signal Process.*, vol. 41, no. 10, pp. 5578-5613, 2022.  
<http://dx.doi.org/10.1007/s00034-022-02036-0>
- [48] X. Feng, H. Chen, G. Wu, A. Zhang, and Z. Zhao, "A new vibration controller design method using reinforcement learning and FIR filters: A numerical and experimental study", *Appl. Sci.*, vol. 12, no. 19, p. 9869, 2022.  
<http://dx.doi.org/10.3390/app12199869>
- [49] J. Yang, H. Yang, X. Yang, and J. Yang, "Optimal design of digital FIR filters based on back propagation neural network", *IEICE Electron. Express*, vol. 20, no. 1, p. 20220491, 2023.  
<http://dx.doi.org/10.1587/ele.19.20220491>

- [50] A. Sridevi, and A. Sathiya, "Integration in CNN and FIR filters for improved computational efficiency in signal processing", *Ain Shams Eng. J.*, vol. 16, no. 1, p. 103201, 2025. <http://dx.doi.org/10.1016/j.asej.2024.103201>
- [51] K.K. Bhadavath, and Z. Mary Livinsa, "Higher order FIR filter architecture design using optimized booth multipliers and SRCSLA adders with retiming for denoising of ECG signals", *Int. J. Commun. Netw. Inf. Secur.*, vol. 16, no. 4, pp. 105-119, 2024.

---

© 2026 The Author(s). Published by Bentham Science Publisher.



This is an open access article distributed under the terms of the Creative Commons Attribution 4.0 International Public License (CC-BY 4.0), a copy of which is available at: <https://creativecommons.org/licenses/by/4.0/legalcode>. This license permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.