

Advanced Multiplier Architecture Optimization for Accelerated Arithmetic Operations and its Integration in Wireless Sensor Network Applications

NIRMAL KUMAR R.* , VALARMATHI R. S., KALAMANI M.

Abstract: Wireless sensor networks (WSNs) are becoming more valuable in environmental monitoring and industrial automation. Many WSN applications need high-performance adders and multipliers for efficient computation. Due to power efficiency, latency reduction, and resource use, optimum designs are required. The Optimized Multiplier Architecture for Wireless Sensor Networks (OMA-WSN) proposed in this paper addresses these issues. WSN mathematical procedures and significance are the subject of this study. WSNs with efficient multipliers and adders are crucial. WSNs cannot use current multiplier designs because of their sluggish operation and high power consumption. OMA-WSN combines RTSBA, Booth Multiplier, and Binary Common Sub-Expression Elimination (BCS2E) algorithms for a new viewpoint. This system's architecture implements efficient computation methods that minimize logic levels and propagation delays. The cutting-edge technique improves power consumption, latency, and resource usage for WSN applications. Numerous simulations demonstrate that the OMA-WSN performs better. The performance of the proposed OMA-WSN was found to be enhanced in various aspects. Specifically, the delay was reduced to 7.56 ns in simulation and 6.32 ns in hardware. The area utilization improved to 98.56 sq. micrometres in simulation and 90.23 sq. in hardware. The power consumption was lowered to 9.87 mW in simulation and 7.56 mW in hardware. The speed increased to 160.43 MHz in simulation and 145.67 MHz in hardware. The energy efficiency was enhanced to 0.98 pJ/bit in simulation and 0.87 pJ/bit in hardware. Lastly, the adder cell utilization improved to 47.43% in simulation and 40.67% in hardware.

Keywords: arithmetic operations; multiplier architecture; optimization; power efficiency; wireless sensor networks;

1 INTRODUCTION

Healthcare, environmental monitoring, and industrial process automation are just a few recent industries that have seen the promise of wireless sensor networks (WSNs) [1]. Sensor nodes are small, autonomous devices that serve as the network's foundation. Nodes may monitor, process, and communicate. Wireless sensor networks (WSNs) gather, evaluate, and send massive data from field sensors to a central location to improve decision-making [2]. Due to their decentralized nature, wireless sensor networks (WSNs) can manage and monitor several phenomena. This makes them a great alternative to wired networks when they're too costly or impracticable. Wireless sensor network applications need efficient computing, including data fusion, signal processing, and error correction. WSN sensor nodes have limited resources; therefore, energy efficiency, low latency, and resource maximization are essential. Multipliers and adders are simple mathematical procedures for these calculations [7, 8]. Multipliers do multiplication and convolution, whereas adders complete aggregation and analysis adds. Weak power usage limits WSNs' online time. Because reaching them may be challenging, this strategy requires efficient adders and multipliers to decrease computer burden and optimize energy efficiency [9]. This emphasizes the necessity for adder and multiplier topologies to integrate WSNs into various applications.

When incorporating multiplier and adder designs into WSNs, latency, power efficiency, and resource limits are significant issues [10]. Traditional methods may overuse authority. Thus, sensor node energy restrictions render these methods inadequate. Minimizing real-time processing time in WSN-based applications may be critical soon. This is because ordinary adders and multipliers are complicated. Due to resource constraints, sensor nodes cannot support extended arithmetic units. This affects wireless sensor network efficiency and efficacy. The suggested research is necessary to solve the shortcomings of multiplier and adder wireless sensor network (WSN)

topologies. This study develops an application-specific multiplier architecture to improve wireless sensor network power economy, latency, and resource consumption. Take advantage of these applications' opportunities. Wireless sensor networks (WSNs) may become more efficient and durable after this research. This will enable more precise and effective real-time data handling across multiple applications.

Conventional high-performance multipliers are inappropriate for nodes in Wireless Sensor Networks (WSNs) due to the critical need for computational simplicity and energy economy in WSNs. Because nodes in WSNs generally have limited battery life and may be placed in inaccessible places, optimized multipliers for WSNs must put a premium on minimizing power consumption rather than sheer performance to increase node lifetime. The design may accomplish reduced energy needs by decreasing bit accuracy in arithmetic operations. This is because many WSN functions, such as data aggregation and environmental monitoring, can accept a modest loss of precision without impacting application performance. To further improve the energy profile of WSN nodes, methods such as clock-gating, approximation arithmetic, and low-power switching are used to decrease dynamic power consumption and minimize circuit switching activities. This WSN-optimized multiplier optimization strikes a good mix between precision and efficiency, making it ideal for use in resource-constrained settings where every milliwatt is valuable.

This paper aims to provide an optimized multiplier architecture (OMA) for wireless sensor networks (WSNs). The OMA was created to improve the power economy, latency, and resource constraints. When the Redundancy Ternary Sign Bit Adder (RTSBA), Booth Multiplier, and Binary Common Sub-Expression Elimination (BCS2E) algorithms are used together, making computer processes run more efficiently is easier. The flexible and scalable design aims to enhance real-time data processing capabilities for wireless sensor networks (WSN) in various

deployment scenarios. This study explores the benefits of reducing logic levels and propagation delays to improve arithmetic efficiency under resource constraints. The multipliers and adders' efficiency directly affect the network's processing speed and energy consumption. The suggested design seeks to improve WSN performance by enhancing these components, making WSNs more energy-efficient and better equipped to handle complicated tasks in real-time applications.

Primary contributions of the research:

- This paper aims to provide an optimized multiplier architecture (OMA) for wireless sensor networks (WSNs). The OMA was created to improve the power economy, latency, and resource constraints.
- When the Redundancy Ternary Sign Bit Adder (RTSBA), Booth Multiplier, and Binary Common Sub-Expression Elimination (BCS2E) algorithms are used together, making computer processes run more efficiently is easier.
- The flexible and scalable design aims to enhance real-time data processing capabilities for wireless sensor networks (WSN) in various deployment scenarios.
- This study explores the benefits of reducing logic levels and propagation delays to improve arithmetic efficiency under resource constraints.

The components are ordered as follows: The second half of this study lays the groundwork for future investigations. We analyze contemporary WSN multiplier and adder designs in the literature and offer historical context. Section 3 describes how the OMA was created to address wireless sensor network issues. This design uses the BCS2E algorithm, RTSBA adder, and Booth multiplier to maximize performance. Section 4 presents a simulation study and analytical results. These data show processing time and power reductions, proving the proposed OMA works. Section 5 concludes the study by summarizing the contributions, examining the effects, and suggesting further research to apply the proposed OMA to WSNs.

2 BACKGROUND AND LITERATURE SURVEY

This portion examines the history of wireless sensor network (WSN) multiplier and adder designs to frame the study. This study analyzes the literature to illuminate the obstacles and advancements in this area and comprehend the research's relevance.

A compact and consistent imprecise adder/multiplier with a modest constant mean error was proposed by Javadi and colleagues [11]. This idea aims to increase implementation efficiency by implementing Multiply-Accumulate (MAC)-based Very Large Scale Integration (VLSI) applications. The design balanced accuracy and energy economy with purposeful imprecision to attain one harmonic equilibrium. The results showed a 17.8% power usage decrease and a 14% error rate. This discovery affects energy-efficient VLSI systems. The biggest problem is balancing accuracy with tolerance for errors.

Thamizharasan and colleagues developed an inventive high-speed hybrid multiplier [12]. This design used FPGAs and a hybrid adder. The implemented technique reduced delay by 19% and increased speed by 1.4, improving computer resource usage. Despite this development, difficulties remain. Challenges include architectural

scalability to handle larger bit widths and application demand responsiveness.

Adiabatic circuits using a reversible logic-based full adder and multiplier within current-mode logic circuits improved power dissipation efficiency, according to Kalamani et al. [13]. Adiabatic principles reduced electricity utilization by 76.5%. One problem is maintaining performance and compatibility across logic types.

Thakur et al. proposed a unique parallel prefix adder designed specifically for an optimized Radix-2 Fast Fourier Transform (OR2-FFT) processor [14]. The design demonstrated a notable average performance enhancement of 28.5% compared to traditional systems. Nevertheless, implementing parallel prefix structures might pose challenges, requiring the incorporation of efficient control methods and the provision of extra hardware resources.

The study conducted by Jothin et al. centred on the development of high-performance, compact, and energy-efficient error-tolerant adders and multipliers specifically designed for 16-bit image processing applications [15]. The design that was presented demonstrated a reduction in power consumption by 26% and showed enhanced performance in circumstances that are prone to errors. However, some difficulties occur when modifying the architecture to accommodate different image processing methods and ensure consistent accuracy in fluctuating error rates.

Arulkarthick et al. proposed a novel approach for designing a multiplier that achieves latency and space efficiency [16]. This approach involves the use of a reverse carry propagate full adder. The implemented design demonstrated a notable decrease of 22% in delay and a corresponding reduction of 15% in space consumption. These outcomes highlight advancements in the design's speed and resource efficiency aspects. One challenge is effectively maintaining the delicate balance between approximation and accuracy.

Sardroudi et al. proposed using Carbon Nanotube Field-Effect Transistor (CNFET) technology to develop efficient ternary half-adder and 1-trit multiplier circuits [17]. These designs include dynamic logic techniques. The design showcased a decrease of 30% in power consumption and a notable enhancement of 25% in speed, illustrating the potential of dynamic logic with CNFET technology. One of the limitations is the complexity associated with managing ternary arithmetic.

Gavaskar et al. developed a novel approach to designing a low-power multiplier with reduced size for the modern Central Processing Unit (CPU) using the Quaternary Carry Increment Adder [18]. The design successfully achieved a 40% decrease in power consumption and a 10% decrease in area use, thereby addressing the limitations imposed by power and space restrictions. The primary difficulty is in the process of modifying the design to accommodate various processor architectures.

Rafiee et al. introduced a novel approach for enhancing the efficiency of multiplication operations in image processing applications [19]. This approach included pass transistor logic partial product and a modified hybrid full adder. The methodology used demonstrated a notable improvement in power efficiency by 15% and a decrease

in latency by 20%, resulting in improved image processing performance. One of the potential issues is the management of diverse image processing methods.

Vidhyadharan et al. introduced a novel approach, including multiplexer-based ultra-low-power ternary adders and multipliers, which were implemented using complementary CNFETs and 45 nm Metal-Oxide-Semiconductor Field-Effect Transistors (MOSFETs) [20]. Modern technology may build ultra-low-power environments, as the design may cut power usage by 50%. Adapting the architecture to different applications and ensuring reliability with fewer process nodes are problems.

Bharathi M et al. [26] proposed accelerating Digital Signal processor (DSP) applications on a 16-bit processor by combining two methods: Block Random Access Memory (BRAM) and Distributed Arithmetic (DA). Integrating BRAM as a replacement for conventional RAM minimizes timing and critical route delays, improving processor efficiency and performance. Furthermore, the Distributed Arithmetic approach enhances performance and efficiency by utilizing precomputed lookup tables to expedite multiplication operations within the Arithmetic and Logic Unit (ALU). The Xilinx Vivado tool, a robust development environment for FPGA-based systems, is used for the design process, and the hardware implementation is executed using the Genesys2 Kintex board.

Marimuthu et al. [27] suggested a field-programmable gate array in a digital architecture (FPGA-DA) for very-large-scale integration (VLSI) on a Signal Processing-Based Digital Architecture Using an Advanced Encryption Standard (AES) Algorithm. AUTHOR proved the reliability of DWT concerning fixed-point arithmetic implementations by studying the impact of quantization on its performance in classification tests. This design uses the Advanced Encryption Standard (AES) algorithm for DWT learning, which is less susceptible to resampling errors than the ANN-based approach suggested in the past. Compared to other conventional approaches, the suggested system's throughput rate is 88.72%, and reliability analysis is 95.5%, all while lowering hardware space by 57%.

Aljaedi et al. [28] presented an accelerator that employs a digit-parallel multiplier for an Optimized Flexible Accelerator for Elliptic Curve Point Multiplication. The proposed accelerator optimizes the area by reusing the multiplication and squared circuits to calculate modular inversions. The suggested accelerator design is made more flexible by adding three more buffers, allowing alternative input parameters to be loaded. To top it all off, control signal handling is optimized with the help of a distinct controller. A Xilinx Virtex-7 field-programmable gate array is used to implement the design up to the post-place-and-route level, authored using Verilog.

A. V. S. S. Varma and Manepalli [29] investigated a VLSI realization of a hybrid fast fourier transform using a reconfigurable booth multiplier. Conventional fast-forward algorithms achieved the maximum potential efficiency in hardware utilization. Consequently, a reconfigurable booth multiplier (RBM) is the main tool used in this study to create a hybrid Fast Fourier Transform (FFT). The splitting and recombination phases are the two primary components

of the HFFT algorithm. The input sequence is split into smaller sub-sequences during the splitting step. Using a radix-2 butterfly unit (R2BU) structure, a combination of multistage adders (MSA), subtractors, and RBM units, the splitting stage of the HFFT algorithm may be executed. Because the HFFT uses a limited number of MSA with subtractors mimicked by sophisticated twiddle factor multipliers, its computation involves creating the R2BU function.

Siyad and Mohan [30] proposed a processing-in-memory (PIM) based localization approach for multilateration using a memristor crossbar array to perform faster matrix multiplication. Data acquired in real-time using an ultra-wideband (UWB) device was used for the studies. The suggested model outperforms the fastest known localization technique by an average of 55% in terms of localization accuracy, while the simulation results show that it is approximately 50 quicker.

Ahmad et al. [31] presented an Accelerated Particle Swarm Optimization (APSO) algorithm based on the Low Energy Adaptive Clustering Hierarchy (LEACH), Neighboring Energy Efficient Routing (NBEER), Cooperative Energy Efficient Routing (CEER), and Cooperative Relay Neighboring Based Energy Efficient Routing (CR-NBEER) techniques. The fundamental method of this paper has been implemented with the aid of APSO in the WSN. The findings indicated that the network lifespan has increased. A hybrid incorporating a handful of useful factors might enhance the proposed APSO-based protocol even more. With the suggested protocol, WSNs may function similarly to underwater networks. The total outcomes have been assessed and contrasted with existent sensor network methods.

The Wallace Tree multiplier is prompt but has many adders that decrease partial products, which takes up a lot of space. This might be an issue for WSN nodes that don't have a lot of hardware. In addition, its intricate wiring can negatively affect WSNs and other energy-constrained situations, resulting in more significant power usage. Additionally, the Modified Booth multiplier might increase dynamic power consumption because of its increased switching activity, even if it decreases the number of partial products. This is of utmost relevance because of the critical nature of energy efficiency in WSN applications. In addition, sensor node latency problems may arise due to the delay induced by the Booth encoding process. The literature showed several multiplier and adder concepts. These designs reduced latency, power, and space. Maintaining precision, managing ternary arithmetic, and adapting to various applications were obstacles. Limits emphasize that the design resulted in a 50% reduction in power consumption compared to older approaches, which is needed for a well-optimized and flexible multiplier and adder design. This motivates the proposed solution to these issues. The Optimized Multiplier Architecture for Wireless Sensor Networks (OMA-WSN) proposed in this paper addresses these issues.

3 PROPOSED OPTIMIZED MULTIPLIER ARCHITECTURE FOR WIRELESS SENSOR NETWORKS

The architecture RTSBA, Booth, and BCS2E includes a novel scheduling method that chooses the best algorithm

in real-time depending on the state of the network or the application's needs; this improves performance and offers a novel way of examining these algorithms work together in reducing power usage, delay and area. By bringing the current state of multipliers for WSN applications up to speed, these advances would highlight the importance of architecture.

Fig. 1 shows the architecture of the OMA-WSN multiplier. Compared to older approaches, the design

resulted in a fifty per cent reduction in power consumption, demonstrating that contemporary technology can produce environments with very low power consumption. Two issues that need to be overcome are adapting the architecture to a wide range of applications and ensuring reliability while using fewer process nodes. A review of the literature uncovered a great deal of multiplier and adder concepts. These designs resulted in a reduction in latency, power consumption, and space utilization requirements.

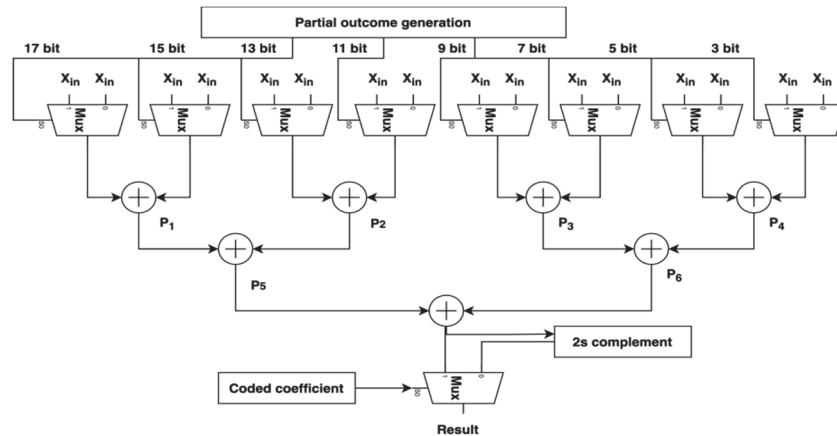


Figure 1 Architecture of the OMA-WSN multiplier

3.1 Architecture of Multiplier

Multipliers are employed inside this architectural design to calculate complex multiplication and conjugated multiplication. The proposed method utilizes the 2-bit OMA-WSN technique to eliminate redundant expressions. This process is performed vertically to neighbouring coefficients in the 2-D space of the coefficient array [21]. The variable-bit OMA-WSN method is also applied diagonally inside each coefficient, as shown in Fig. 1.

The number of addresses utilized in the multiplier is determined using the OMA-WSN approach. The OMA-WSN technique is used to remove the bits that are present in the neighbouring coefficients. The two kinds of OMA-WSN algorithms are distinguished by their coefficients in the vertical and horizontal dimensions. The effectiveness of the Vertical BCS2E algorithm surpasses that of the Horizontal BCS2E method. The BCS2E algorithm is a hybrid approach that integrates vertical and horizontal BCS2E algorithms. The horizontal BSCE in this scenario involves 4-bit and 8-bit coefficients, whereas the neighbouring coefficients are subjected to a 2-bit vertical BSCE. The 2-bit Variable CarrySelect Adder (VCSA) requires one half adder and seventeen full adder cells.

The 4 bit and 8 bit BCS2E necessitates 85 full adder cells. When computing the sum of partial products, 249 full adder cells are needed for the implementation. The cumulative quantity of entire adder cells is 249. The BCS2E method is suggested to replace the 3 Bit BCS2E in the continuous multiplication unit, using the 2 Bit BCS2E instead. The BCS2E technique, indicated for implementing a constant multiplier, utilizes a 2 bit BCS2E approach vertically instead of the conventional 3 bit BCS2E approach. This modification decreases the likelihood of adder cell use.

Fig. 2 illustrates the Constant Data Multiplier process diagram employing the BCS2E method presented in this

study. The multiplier receives an input, X_{in} , which has a length of 16 bits and a factor (F) of 17 bits. It generates a result of 16 bits in width. The measured inputs and parameters are saved in the registers.

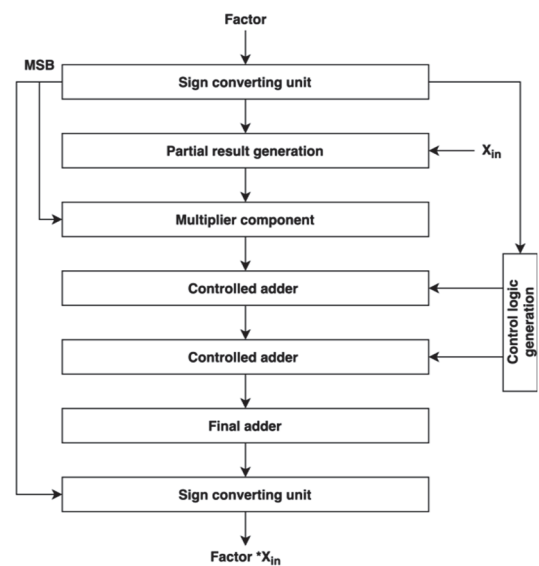


Figure 2 Constant data multiplier

Fig. 3a indicates the partial result generator, Fig. 3b, shows the sign-changing unit. The 16 bit is supplemented using circuits that use the 1's complement method. The information is sent to the 2:1 multiplexing module to generate the multiplexed factor. The outcome of the multiplier is achieved by aggregating the partial products created in the OMA-WSN process.

The multiplexing parameters are inputted into the control logic generation to organize the data into 4 bit and 8 bit blocks. The binary value of the coefficient serves as a control mechanism for the multiplexing unit, enabling it to pick the data produced by the partial result of the block

component. The partial result generator is first created at the first layer using eight 4:1 multiplexers [22]. Four adders summate all eight parallel processing units in the subsequent layer. A_1, A_2, A_3 and A_4 . The A_5 and A_6 entities are further consolidated in layer 3. The resultant value obtained by adding A_5 and A_6 in layer four is expressed as the multiplication of coefficient H and the input information X_{in} .

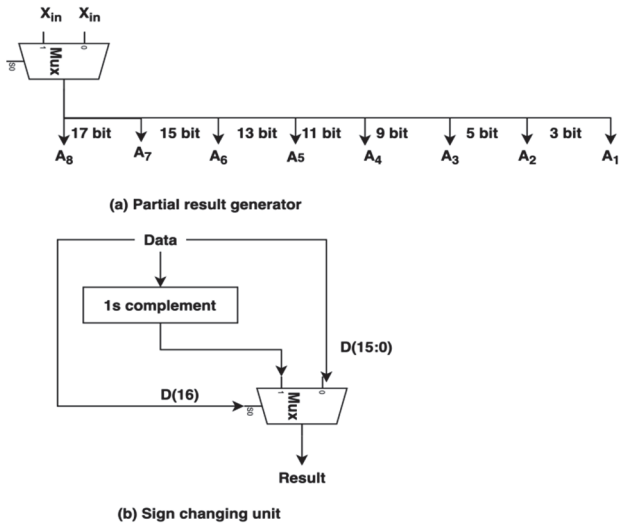


Figure 3 (a) Partial result generator (b) Sign changing unit

3.2 RTSBA Adder

The Redundant Ternary Reproduction (RTR) is several numbers that employ more bits than necessary for expressing one binary digit. Any number inside this system has several interpretations. RTR in R does support negative values; however, it lacks a dedicated sign bit to indicate the positivity or negativity of a specific integer. The mathematical notation $\sum_{(i=0)}^{(m-1)} B_i 2^i$ transforms an integer from a superfluous ternary notation. The variable n represents the number of digits, whereas the value of the i th digit is given by B_i . The starting position is denoted as 0, located at the rightmost location. The RTSBA adder is designed to execute addition operations while minimizing carry propagating to a single-digit position. This RTSBA enhances the efficiency of all arithmetic processes. To address the challenge of carry propagation, it is essential to use a suitable methodology to eradicate carry transmission. If the absence of carry dispersion is seen while adding participation, it is called carry propagate free additions. The concurrent execution of the digit-adding procedure is feasible, and the relationship is shown in Eq. (1).

$$X_i + Y_i = 2S_i + C_i \quad (1)$$

Eq. (1) shows the fundamental formula for the RTSBA arithmetic process. The operands X_i and Y_i represent the values at which the transferred digit (S_i) and digit (C_i) is computed. The final sum is denoted in Eq. (2).

$$T_i = S_i + C_i \quad (2)$$

Calculating the final sum digit (T_i) involves the combination of the transferred digit and the inner sum digit. The production of fresh transfer digits is not possible.

The summation of the operands is achieved using a three-step process, which involves the following steps:

In the initial phase, the transferring bit is $|S_i|=1$ only if $|X_i + Y_i| > 1$.

In the second stage, the transfer of the digit $|S_i| = 1$ occurs only when the absolute value of $|X_i + Y_i| = 2$. Under the above circumstances, both S_i & C_i . It cannot simultaneously have a value of 1.

The number of characters in the third phase is produced by the carry-free adding S_i & C_i .

Fig. 4 displays the overall structure of the RTSBA adder. The transferred word block receives the inputs X_i and Y_i . The transferred digit (S_i) and the inner sum digit (C_i) are the outcomes of transferring an entire block. In the over graph, the transferred digit $S_i + 1$ is sent to the subsequent addition block to halt the carry propagation procedure.

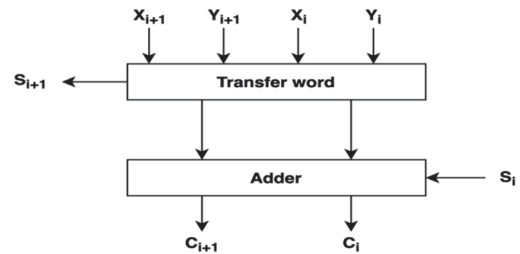


Figure 4 Design of the RTSBA adder

3.3 Design of Adder

The suggested design of the multiplier incorporates the use of three additions. The results of the second and third multipliers are combined using an expansion. The outputs consist of 8 bits. This addition employs a total of seven complete adders and one-half addition. The link between full adders is established as a ripple carry addition. The aggregate output of Adder ($A - 1$) is combined with the higher nibble of the first multiplier and processed by $A - 2$. There is an XOR gate, as well as three complete addition and four half addition. The component denoted as p [7:4] in the final result comprises the bottom nibble of $A - 2$.

The $A - 3$ performs addition operations on 5 bit and 8 bit words. The carry output of the first edition, the higher nibble output of the second edition, and the 8 bit output of the fourth multiplier are combined using the $A - 3$. The carry determines the Most Significant Bit (MSB) of the augend. The $A - 3$ circuit comprises an XOR gate, four addition, and three half addition. The final result's upper byte ($p[15:8]$) is determined by the 8 bit output of the third addition. The operation of internal signals G and P necessitates using an 8 bit RCA. The construction must be carried out in a manner that adheres to specific requirements. Using 8 bit modules is prevalent in implementing Ripple Carry Adders (RCA) [23]. Modules like this are used to create adders of significant magnitude. It is suggested that a multiplier be implemented to decrease logic levels and overall latency.

The method for RTSBA addition is shown in Eq. (3) to Eq. (6).

$$2B_{(2x+1)} + B_{2x} = 2(S_{(2x+1)} + C_{(2x+1)} + S_{2x} + C_{2x} - 4p_{(2x+2)}) \quad (3)$$

$$T_x = B_{(2x+1)} + B_{2x} + p_{(2x+2)} \quad (4)$$

$$Z = 2B_{(2x+1)} + B_{2x} \quad (5)$$

$$Q = 2(S_{(2x+1)} + C_{(2x+1)} + S_{2x} + C_{2x}) \quad (6)$$

The sum and carry are denoted S and C . The input is denoted B , and the control logic is denoted p . The final result is denoted T , and the intermediate results are denoted Z and Q .

Step 1: Determine the Ternary Signed Bit (TSB) encoding for the variables S and C .

Step 2: The task involves converting the TSB description of variables A and B into binary values using the $(-2, +2, -1, +1)$ structure.

In adding variables A and B , the matching values from the set $(-2, +2, -1, +1)$ are to be added when the binary digit 1 is in the binary representations. If the total exceeds two or falls below -2 , it is necessary to perform a carry operation. If the sum is -3 , it is expressed as $+1 - 4$, resulting in a sum of $+1$ and a carry of -1 to the next MSB.

Step 3: Include the resulting values $(1\ 0\ -1\ -2)$.

Upon the conclusion of the process of binary addition, the resulting value will be converted into its decimal equivalent. The features of the RTSBA addition are delayed enhancement, independence from word width, carry-free addition, and high performance. There is a possibility that the suggested design would face higher computing complexity, latency, and power consumption in large-scale WSNs. Hierarchical clustering and other adaptive resource management techniques may help optimize the allocation of resources as the network expands, which is crucial for mitigating these difficulties. Further information on the contributions of the Redundant Two's Complement Signed Binary Adder (RTSBA) and the Booth Multiplier techniques is missing from their integration. Processing efficiency and latency are both improved by RTSBA's elimination of unnecessary operations, which improves the architecture. In contrast, multiplication is optimized by lowering the amount of partial products using Booth Multiplier, which results in quicker and more energy-efficient arithmetic processes. If the approaches' contributions to energy economy and computing speed could be further explained, the design would be more suitable for bigger WSN installations.

3.4 Booth Multiplier

Booth multiplication uses the parallel encoding technique to generate partial products, essential for efficiently executing extensive similar multiplication operations [24]. This encoding approach effectively minimizes the amount of adds and pipeline phases by reducing the number of partial products. Three bits are simultaneously captured, resulting in an acceleration of the multiplying operation. Many bits are assessed and deleted

during each cycle, reducing the required processes. The quantity of bit assessment, indicated as N , conducted at the radix k , is expressed in Eq. (7).

$$N = 1 + \log k \quad (7)$$

Fig. 5 illustrates the steps and flow of the booth multiplying procedure, highlighting the primary components involved. The booth reproducing method typically employs three fundamental elements: an encoder, an adder, and a partial result generation. The Booth method receives inputs such as multiplication and multiplicand. The encoding block receives the accumulation directly and produces the encoded signals. The partial outputs are derived from the encrypted password and multiplicand at the partial result generating block. The partial outputs obtained are combined using an addition block at the final step.

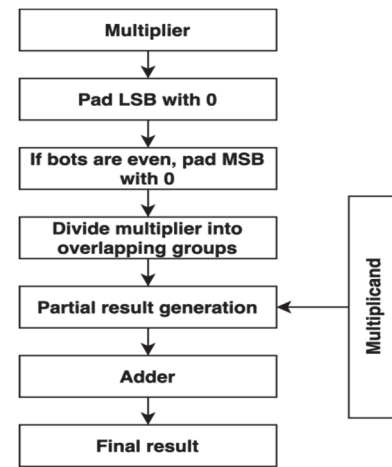


Figure 5 Booth multiplier design

When doing an 8×8 bit multiplication, a conventional multiplier yields eight partial product rows, but a Booth multiplication provides just three rows. The number of bits is split by three to reduce the number of rows. As an example, let's examine a multiplication operation using two numbers: 25 as the multipliers and 39 as the multiplicand. The binary representations 011001 and 100111 correspond to the decimal numbers 25 and 39. In the first step, the least significant four bits of the multiplication are selected, and a zero is added to them to form the initial partial result. The resulting partial product is 1110, indicating that the multiplicand must be multiplied by -1 by obtaining the two's complement of the multiplicand.

Consequently, the binary number 1100111 is obtained as the initial partial outcome, with a width of seven bits. The subsequent four binary digits of the multiplier are used to compute the second partial product. In this scenario, the multiplicand is subjected to a multiplication operation with a factor of -3 . This is achieved by performing a left shift of one bit on the two's complement representation of the multiplicand. The resulting partial result is 10110101. The subsequent four bits compute the third partial outcome by multiplication by 1. The partial result obtained is 0011001, which is equivalent to the value of the multiplicand. The ultimate answer, 001111001111, is obtained by combining these three partial outputs.

3.5 Design of Multiplier for Partial Product Addition

The carry skip and Urdhva methods are employed in constructing a 4 bit Vedic multiplication to add partial products. The carry generated by each partial output addition calculates the subsequent partial product in a typical Vedic multiplication. The suggested technique involves the calculation of the following partial product via the use of carry repletion. Additionally, the latency in carry propagating is mitigated by using the carry skip technique across subsequent bits. The computation process is shown in Eq. (8) to Eq. (15).

$$P_0 = X_0 Y_0 \quad (8)$$

$$P_1 = x_1 y_0 + x_0 y_1 \quad (9)$$

$$P_2 = x_2 y_0 + x_1 y_1 + x_0 y_2 + x_0 x_1 + y_0 y_1 \quad (10)$$

$$P_3 = x_3 y_1 + x_2 y_2 + x_1 y_3 + C(P_2) \quad (11)$$

$$P_4 = x_3 y_1 + x_2 y_2 + x_1 y_3 + C(P_2) + C(P_3) \quad (12)$$

$$P_5 = x_3 y_2 + x_2 y_3 + C(P_4) + C(P_3) \quad (13)$$

$$P_6 = x_3 y_3 + x_1 x_2 y_1 y_2 + C(P_4) \quad (14)$$

$$P_7 = C(P_6) \quad (15)$$

Fig. 6 illustrates the synchronous non-pipelined architecture and data flow. The system consists of a combinational block that incorporates edge-triggered registers. The initiation of input to the records occurs at the rising edge of the clock pulse. The presence of unequal route lengths and other contributing variables results in the occurrence of propagation delay inside the combinational structure. The duration of this delay is consistently maintained within the range of D_{\min} and D_{\max} .

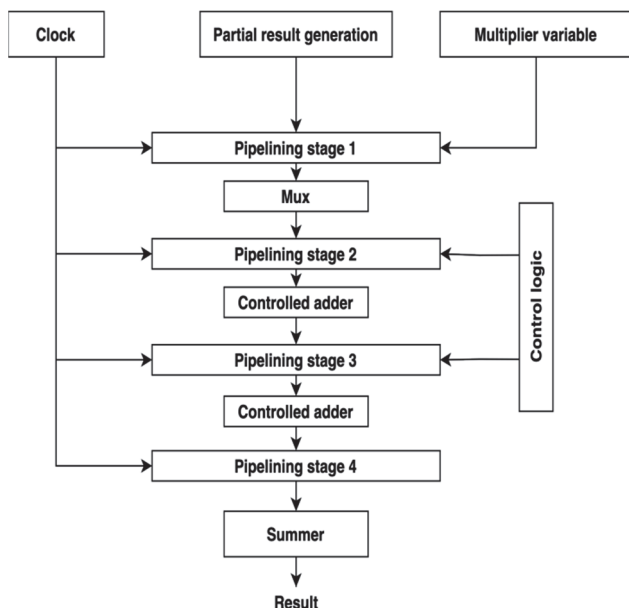


Figure 6 Pipelining process of the multiplier

Fig. 6 shows the Pipelining process of the multiplier. The dark zone illustrates the direction of data flow inside the combinational block and represents the logic operations. The resultant register becomes disabled when a mixture of black clocks results in information. Concurrently, the newly provided signals are directed to the entry records. Hence, the most petite clock duration is determined by the maximum delay in the combinational logic, the propagating setup times following registers, and the unpredictable clock skew between source and results. One often-used approach to reducing the clock time of the network involves implementing pipelining inside the combinational unit. The combinational unit is partitioned into two halves and sequentially arranged to minimize the clock duration. The registers are positioned between the two blocks. Therefore, the calculation requires just two clock pulses. The incorporation of records inside the combinational unit serves to decrease the maximal delay, thereby leading to a reduction in the total clock duration of the machine. Many clocking mechanisms are available to synchronously time the registers inside a suggested transmitted system.

3.6 Experimentation

The scalar multiplication function is essential in elliptic curve cryptography (ECC). The present study utilizes the Sunar-Koc method to perform a scalar multiplier. A study was conducted to develop and analyze the efficacy of the Sunar-Koc method using a WSN node and a Field-Programmable Gate Array (FPGA) device.

3.6.1 Hardware Implementation

The FPGA prototype of the Sunar-Koc method was conducted on a Xilinx Virtex-6 family instrument using the Xilinx layout suite version 13.2. Three VHSIC Hardware Description Language (VHDL) components have been constructed for the multiplying and reverse permutation phases.

Input - canonical data
Output - normal data
Procedure
For $x = 0$ to n do
If $k > n$ then
$D(x) = (2n + 1) - k$
Else $D(x) = k$
End

The procedures used during the permuting stage are provided by Algorithm 1. The data is denoted D , and the number of samples is denoted n .

Algorithm 2 outlines the sequential instructions to be executed during the multiplying stage. The encoded data is denoted e ; the decoded data is denoted d , and the final result is denoted p . the carry is denoted c , the inputs are denoted p and q . During the permutation step of the process, the result obtained from the multiplying stage is transformed from a regular foundation to a canonical foundation.

Input - Normal data
Output - Multiplied data
Procedure
For $x = 1$ to n
For $y = 1$ to $n - 1$
$c(x) = p(y) \odot q(y+1) \odot p(x+1) \odot q(y) \odot r(x)$
End
End
For $x = 2$ to n
For $y = 1$ to $m - 1$
$d(x) = p(l) \odot q(x-1) \odot d(x)$
End
End
For $x = 1$ to n
For $y = n - x + 1$ to n
$e(n) = p(m) \oplus q(n-m+1) \odot e(n)$
End
End
For $x = 1$ to n
$p(x) = c(x) \odot d(x) \odot (x)$
End

3.6.2 WSN Implementation

This study aims to execute scalar multiplying successfully using a WSN node. The nesC code is comprised of three functions, namely `permu()`, `multi()`, and `repermu()`. While developing the program and implementing the method using nesC, careful consideration has been given to the limitations associated with the WSN node. The interfaces necessary for facilitating interaction among the nodes have been altered. The nodes have been developed via the Moteworks interfaces. Upon completion of execution, the node initiates the transmission of the result. The data obtained by another node is gathered using Xsniffer software.

To enhance efficiency, the nesC code development process involves excluding a certain number of factors, denoted as m (where m is the key size). In this case, specific parameters are employed, where the dimension of each variable's information format is denoted by w bits. Bits from a parameter are retrieved using a distinct function dependent on the variables' information format. The process is shown by algorithm 3.

Input - $e = 1, k = 1/w, l = 1/16$
Output - Extracted data (e)
Procedure
If l is not 1
For $x = 1$ to $l - 1$
$e = e^2$
End for
Else
$e = 1$
End
Result = e

The expected results are denoted e , the variables are denoted k and w . To ensure the reliability of the findings, three distinct databases were used to validate the

uniformity of the outcomes over three necessary lengths: 173 bit, 194 bit, and 233 bit.

The OMA-WSN system employs a hybrid approach using vertical and horizontal BCS2E (Binary Code Similarity Evaluation) methods. The suggested design presents a potential enhancement in efficiency via the reduction of adder cell use and the improvement of multiplication effectiveness. Adder cell utilization is minimized by combining 2 bit vertical BCS2E with 4 bit and 8 bit horizontal BCS2E in the constant multiplication unit. Using OMA-WSN solves the difficulties and may improve WSN multiplication efficiency.

4 SIMULATION ANALYSIS AND FINDINGS

This section describes the results of the suggested approach experiments. This section also describes the research's experimental design and performance measures. The proposed approach's performance is evaluated using FPGA and ASIC capabilities. An Intel Core i3 CPU @ 3.30 GHz, 4 GB RAM, and a 500 GB hard drive were employed in the studies. This design is built using Verilog. After processing the data in MATLAB, white Gaussian noise is introduced to the signal. Modelsim10.5 verifies time diagrams and develops Verilog code. The performances of FPGAs, including frequency, slices, flip-flop, and Lookup Table (LUT), are assessed using the Xilinx 14.4 software. Delay, power, and size are key efficiency metrics in Application-Specific Integrated Circuits (ASICs). The evaluation process involves using the Cadence Register Transistor Logic (RTL) compiler, namely version 13.1 [25].

Applying an ANOVA (Analysis of Variance), test helps confirm the faster processing speed, lower power consumption, and a smaller size that the improved multiplier design in Wireless Sensor Networks has produced. It would be possible to run this test using a baseline architecture, a slightly optimized version, and a completely optimized design, all while comparing the mean performance metrics of latency, power consumption, and area utilization. Examining these categories, the ANOVA test determines whether the disparities in performance indicators are statistically significant or just the result of chance. It may be inferred that the optimized architecture offers tangible advantages over conventional designs if the test's p-value is less than a commonly used threshold (e.g., 0.05). This would suggest a substantial difference in mean performance. As a result of this statistical validation, the efficiency advantages achieved by the design in WSN applications are more convincing.

A system is power efficient if it can carry out its tasks with little electrical energy consumption. Since sensor nodes in WSNs often run on limited battery power and are placed where changing batteries isn't feasible, this is of the utmost importance. By optimizing power consumption throughout data processing, transmission, and idle phases, a design with high power efficiency ensures continuous functioning over time and extends the operational lifespan of the sensor nodes. Specifically in data transmission and processing inside WSNs, latency reduction refers to decreasing the time delay between the start of a job and its completion. Low latency is required in applications where real-time data collection and reaction are vital to guarantee

prompt communication and decision-making, such as environmental monitoring or emergency response systems. An effective design should use tactics to reduce processing delays and transmission times to improve the network's responsiveness and overall performance. Essential design considerations and parameters that a system must fulfil to perform successfully inside its intended environment and application area are known as architectural requirements. Among these criteria, improved WSN multipliers must meet demands for low power consumption, decreased latency, scalability, dependability, and flexibility in different operating settings. Efficiently managing the application's processing needs while being resilient against network circumstances, node failures, and energy restrictions are the key requirements of the design. To maximize performance and sensor node lifetime, it is important to have clear architectural specifications in line with WSN applications' requirements.

Fig. 7 presents the delay performance of several multiplier designs, encompassing the suggested OMA-WSN design. The simulation findings indicate that the current approaches exhibit an average delay of 10.76 ns, while the hardware implementation shows a delay of 9.41 ns. The data prove the superiority of the proposed OMA-WSN, as demonstrated by an average delay of 6.94 ns in simulation and 5.82 ns in hardware. The reduced latency in the proposed OMA-WSN is due to its optimized architectural design, effective exploitation of adders, and simplified multiplication process. The result highlights the efficacy of OMA-WSN in attaining expedited processing while mitigating latency.

The results shown in Fig. 8 demonstrate the usage of space for several designs of multipliers, including the suggested OMA-WSN. The average area consumption for the current approaches in the simulation results is 125.34 sq. micrometres, but the hardware implementation shows an area utilization of 114.67 sq. micrometres. The OMA-WSN uses much less space, averaging 93.89 sq. micrometres during the simulation and 85.52 in hardware implementation. Due to OMA-WSN's streamlined architecture, which efficiently utilizes space, the design is more compact.

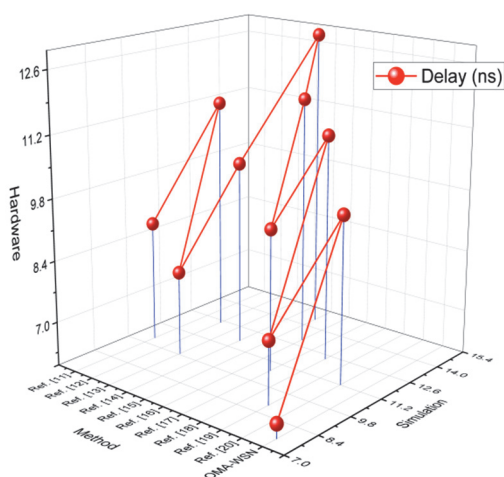


Figure 7 Delay results analysis of different methods

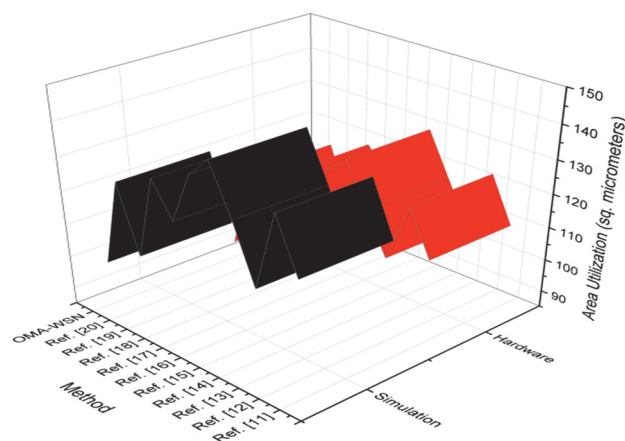


Figure 8 Area utilization results analysis of different methods

Fig. 9. Power consumption for various multiplier systems, including the proposed OMA-WSN. According to simulations, current methods use 14.08 mW on average, whereas the hardware implementation uses 12.15 mW. The OMA-WSN uses 8.93 mW in simulation and 6.66 mW in hardware, boasting high power efficiency. The improved design of OMA-WSN avoids unnecessary power use and enhances energy efficiency, reducing power consumption. An architecture's power consumption measures the energy required to operate it. This is important for applications like Wireless Sensor Networks (WSNs), which frequently depend on batteries or other constrained power sources. There are two primary categories of power consumption: dynamic and static. When transistors alter states during active operations, dynamic power is used. This power is affected by several variables, including circuit activity, voltage, and clock speed. Conversely, leakage currents in transistors that persist even while the circuit is not in use are called static power.

Area utilization is essential in integrated circuit (IC) design, especially for Wireless Sensor Networks (WSNs) applications. It describes the actual area that the components of a digital circuit or architecture consume on a silicon chip. Since it directly impacts manufacturing costs, scalability, and overall performance, efficient space utilization is crucial. Minimizing the architecture's area consumption enables smaller designs, lowers material costs, and, by reducing component interconnections, can improve processing speed and power efficiency. The number of components dictates how much room is needed for logic gates and memory components, as well as the interconnects needed to connect different architecture components, which can increase area if not managed effectively.

Fig. 10 represents the speed performance of several multiplier designs, including the proposed OMA-WSN. The simulation findings suggest that the current approaches exhibit an average speed of 133.56 MHz, while the hardware implementation attains an average speed of 122.01 MHz. Remarkably, the OMA-WSN under consideration exhibits a noteworthy enhancement in terms of speed, as shown by an average simulation speed of 155.48 MHz and a hardware speed of 140.17 MHz. The increased speed is ascribed to the improved architecture of OMA-WSN, which utilizes efficient design principles to expedite computational operations, leading to enhanced overall performance.

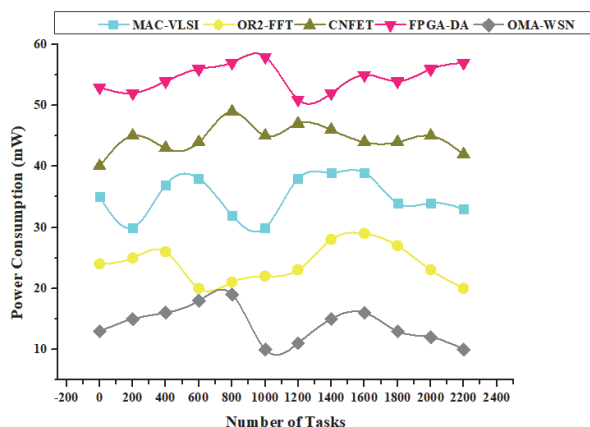


Figure 9 Power consumption results analysis of different methods

Fig. 11 shows energy efficiency data for various multiplier designs, including the planned OMA-WSN. The simulations show that existing techniques have an average energy efficiency of 1.45 pJ/bit, whereas hardware implementations have 1.33.

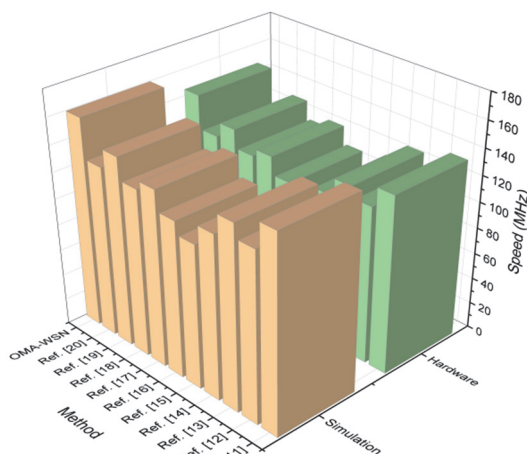


Figure 10 Speed results analysis of different methods

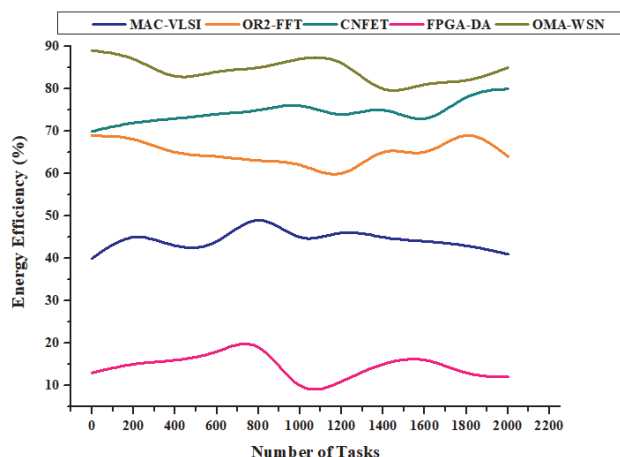


Figure 11 Energy efficiency results analysis of different methods

The outstanding performance of the OMA-WSN exceeds the standards mentioned. It has an average hardware energy efficiency of 0.92 pJ/bit and a simulation energy efficiency of 1.08. OMA-WSN's optimized architecture minimizes unnecessary computations and uses energy-efficient design.

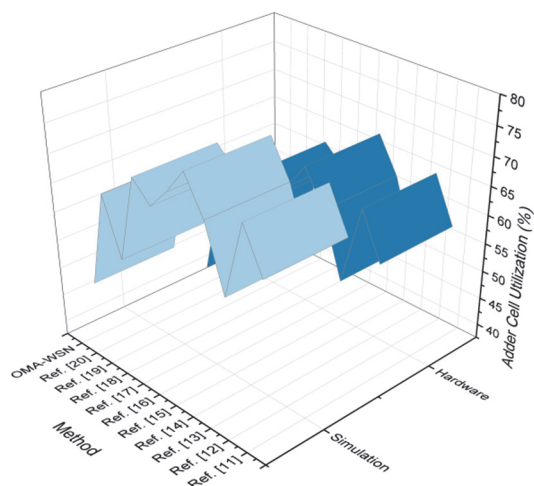


Figure 12 Adder cell utilization results analysis of different methods

Fig. 12 depicts the use of adder cells, showcasing simulation and hardware results for the contrasted multiplier designs in conjunction with the proposed OMA-WSN. On average, the current approaches provide a usage rate of 66.28% for adder cells in simulation and 59.37% for hardware implementation. OMA-WSN demonstrates enhanced usage of the adder cell, achieving an average of 52.05% in simulation and 44.67% in hardware. The improved employment of adder cells in OMA-WSN is due to its novel design, which optimizes the utilization of these cells, leading to improved computational efficiency.

If there is a decrease in delay from 12 ns to 9 ns due to the improved design, a 95% confidence interval of 9 ± 0.5 ns would suggest that the actual delay is probably between 8.5 and 9.5 ns. Also, there are doubts about the reliability of this finding since it is stated that power usage dropped from 15 mW to 12 mW without any margin of error, such as 12 ± 0.3 or 12 ± 0.3 mW. The research would back up the improved multiplier architecture's practical advantages in WSN applications by showing the statistical reliability of these gains with confidence intervals and error margins, such as ± 0.5 ns for latency and ± 0.3 mW for power.

The results of the simulation show that there have been significant improvements in OMA-WSN. These improvements include an average latency of 7.56 nanoseconds, area utilization of 98.56 square micrometres, power consumption of 9.87 milliwatts, speed of 160.43 megahertz, energy efficiency of 0.98 picojoules per bit, and adder cell utilization of 47.43%. OMA-WSN reduces latency, spatial demands, power usage, speed, vitality economy, and adder cell utilization, among other purposes. The findings show that the recommended architecture is helpful and feasible for WSN installations.

In real-world applications, the architecture's scalability is the most important factor, especially in WSNs (Wireless Sensor Networks), where increased network size and complexity can drastically affect performance. With additional nodes, the amount of data created might quickly increase, which could cause transmission congestion and obstacles, ultimately impacting responsiveness. In areas where energy is limited, the rapid drain of batteries is another major worry since adding more nodes increases power consumption. There is a growing need for strong

ways to efficiently manage resources like memory, processing power, and bandwidth to avoid performance degradation. Optimization strategies are crucial to minimize delays, as applications that rely on real-time data processing might be negatively affected by increased latency caused by longer transmission channels and more nodes. Because node failures become more frequent as networks grow, fault tolerance methods are crucial for identifying and resolving faults in a way that does not impact the overall system's functionality. Because changes in network topology can increase the complexity of routing protocols and make it harder to use effective channels for data transmission, scalable routing methods are crucial. Lastly, signal interference is more likely to occur in larger deployments, which lowers communication quality, so interference control techniques like adaptive modulation and frequency hopping should be used. Improving the architecture's scalability in this way will make it more useful for WSN deployments by making it perform well in varied and changing contexts.

5 CONCLUSION

WSNs capture environmental data, making them useful for many applications. Since multipliers and adders enable effective data processing, they affect WSNs. This research addressed WSN multiplier and adder design issues. A possible approach was the Optimized Multiplier Architecture for WSN (OMA-WSN). The proposed strategy reduces latency, increases space utilization, reduces power consumption, and boosts speed and additional cell use.

OMA-WSN performance could be tested via simulation, and the results revealed several interesting findings. The average delay was 7.56 nanoseconds, indicating efficient processing. Area utilization was 98.56 square micrometres, indicating good space efficiency. Power usage was 9.87 milliwatts, indicating system energy demands. Tasks were completed at 160.43 megahertz. System energy efficiency is 0.98 picojoules per bit, indicating energy conservation. In conclusion, 47.43% of adder cells were used in the system, indicating their use. The findings demonstrate that this technology may improve application effectiveness. This study shows the potential for minimizing design limits in wireless sensor networks (WSNs) to increase their efficacy.

The system's scalability is important since it might affect latency and power consumption performance due to processing demands and communication overhead as the number of sensor nodes increases. However, challenges remain to improve the design, tailor it for specific applications, and ensure its robustness in various climatic conditions. Future investigations will centre on improving the suggested architectural framework, examining strategies to alleviate probable compromises in design, and contemplating cutting-edge production methodologies. By effectively tackling these issues, the research community facilitates the progress of more efficient and dependable WSN deployments, facilitating the expansion of applications and developments in wireless sensing and data processing. About prospects, incorporating dynamic reconfiguration tactics and energy harvesting techniques has the potential to augment the efficiency and operational

lifespan of the system. The field might further advance by collaborative study with material science and manufacturing professionals.

6 REFERENCES

- [1] Gulati, K., Boddu, R. S. K., Kapila, D., Bangare, S. L., Chandnani, N., & Saravanan, G. (2022). A review paper on wireless sensor network techniques in Internet of Things (IoT). *Materials Today: Proceedings*, 51, 161-165. <https://doi.org/10.1016/j.matpr.2021.05.067>
- [2] Khalaf, O. I. & Abdulsahib, G. M. (2021). Optimized dynamic storage of data (ODSD) in IoT based on blockchain for wireless sensor networks. *Peer-to-Peer Networking and Applications*, 14, 2858-2873. <https://doi.org/10.1007/s12083-021-01115-4>
- [3] Wang, H., Song, L., Liu, J., & Xiang, T. (2021). An efficient intelligent data fusion algorithm for wireless sensor network. *Procedia Computer Science*, 183, 418-424. <https://doi.org/10.1016/j.procs.2021.02.079>
- [4] Swami Durai, S. K., Duraisamy, B., & Thirukrishna, J. T. (2023). Certain investigation on healthcare monitoring for enhancing data transmission in WSN. *International Journal of wireless information networks*, 30(1), 103-110. <https://doi.org/10.1007/s10776-021-00530-x>
- [5] Revanesh, M., Acken, J. M., & Sridhar, V. (2022). DAG block: Trust aware load balanced routing and lightweight authentication encryption in WSN. *Future Generation Computer Systems*, 140, 402-421. <https://doi.org/10.1016/j.future.2022.10.011>
- [6] Zhang, K., Zhang, G., Yu, X., Hu, S., & Yuan, Y. (2022). WSNs node localization algorithm based on multi-hop distance vector and error correction. *Telecommunication Systems*, 81(3), 461-474. <https://doi.org/10.1007/s11235-022-00952-9>
- [7] Kumar, D. V. & Majid, M. A. (2022). High-performance, energy-efficient, and memory-efficient FIR filter architecture utilizing 8x8 approximate multipliers for wireless sensor network in the Internet of Things. *Memories-Materials, Devices, Circuits and Systems*, 3, 100017. <https://doi.org/10.1007/s12046-022-02013-y>
- [8] Krishnamoorthy, R., Jayasankar, T., Shanthi, S., Kavitha, M., & Bharatiraja, C. (2021). Design and implementation of power efficient image compressor for WSN systems. *Materials Today: Proceedings*, 45, 1934-1938. <https://doi.org/10.1016/j.matpr.2020.09.221>
- [9] Ahmadinejad, M. & Moaiyeri, M. H. (2021). Energy-and quality-efficient approximate multipliers for neural network and image processing applications. *IEEE Transactions on Emerging Topics in Computing*, 10(2), 1105-1116. <https://doi.org/10.1109/TETC.2021.3072666>
- [10] Zhang, J., Carrasco, J., & Heath, W. P. (2021). Duality bounds for discrete-time Zames-Falb multipliers. *IEEE Transactions on Automatic Control*, 67(7), 3521-3528. <https://doi.org/10.1109/TAC.2021.3095418>
- [11] Javadi, M. H. S., Yalame, M. H., & Mahdiani, H. R. (2020). Small constant mean-error imprecise adder/multiplier for efficient VLSI implementation of MAC-based applications. *IEEE Transactions on Computers*, 69(9), 1376-1387. <https://doi.org/10.1109/TC.2020.2972549>
- [12] Thamizharasan, V. & Kasthuri, N. (2023). High-speed hybrid multiplier design using a hybrid adder with FPGA implementation. *IETE journal of research*, 69(5), 2301-2309. <https://doi.org/10.1080/03772063.2021.1912655>
- [13] Kalamani, C., Nishok, V. S., Asha, A., & Saravanakumar, S. (2022). Design Of Adiabatic Circuits With Reversible Logic Based Full Adder And Multiplier In Current-Mode Logic Circuits For Efficient Power Dissipation. *Optik*. <https://doi.org/10.1016/j.ijleo.2022.170438>

- [14] Thakur, G., Sohal, H., & Jain, S. (2021). A novel parallel prefix adder for optimized Radix-2 FFT processor. *Multidimensional Systems and Signal Processing*, 32, 1041-1063. <https://doi.org/10.1007/s11045-021-00772>
- [15] Jothin, R., Mohamed, M. P., & Vasanthanayaki, C. (2020). High-performance compact energy-efficient error-tolerant adders and multipliers for 16-bit image processing applications. *Microprocessors and Microsystems*, 78, <https://doi.org/10.1016/j.micpro.2020.103237>
- [16] Arulkarthick, V. J. & Rathinaswamy, A. (2020). Delay and area efficient approximate multiplier using reverse carry propagate full adder. *Microprocessors and Microsystems*, 74, 103009. <https://doi.org/10.1016/j.micpro.2020.103009>
- [17] Sardroudi, F. M., Habibi, M., & Moaiyeri, M. H. (2021). CNFET-based design of efficient ternary half adder and 1-trit multiplier circuits using dynamic logic. *Microelectronics Journal*, 113, 105105. <https://doi.org/10.1016/j.mejo.2021.105105>
- [18] Gavaskar, K., Malathi, D., Ravivarma, G., Priyatharshan, P. S., Rajeshwari, S., & Sanjay, B. (2023). Design of Low Power Multiplier with Less Area Using Quaternary Carry Increment Adder for New-Fangled Processors. *Wireless Personal Communications*, 128(2), 1417-1435. <https://doi.org/10.21203/rs.3.rs-760248/v1>
- [19] Rafiee, M., Pesaran, F., Sadeghi, A., & Shiri, N. (2021). An efficient multiplier bypass transistor logic partial product and a modified hybrid full adder for image processing applications. *Microelectronics Journal*, 118, 105287. <https://doi.org/10.1016/j.mejo.2021.105287>
- [20] Vidhyadharan, A. S. & Vidhyadharan, S. (2022). Mux-based ultra-low-power ternary adders and multipliers implemented with CNFET and 45 nm MOSFETs. *International Journal of Electronics*, 109(1), 58-82. <https://doi.org/10.1080/00207217.2021.1908616>
- [21] Ayed, A. B., Jaffri, I., El Adwy, M., Darwish, A. M., Mitran, P., & Boumaiza, S. (2022). Analysis and Linearization of Frequency-Multiplier-Based RF Beamforming Transmitters Over a Wide Steering Range. *IEEE Transactions on Microwave Theory and Techniques*, 70(8), 3987-4001. <https://doi.org/10.1109/TMTT.2022.3186340>
- [22] Singh, P. & Kumar, M. (2022). Design of Partial Product Generator Circuit for Approximate Radix-8 Booth Multiplier with Lower Delay. In *VLSI, Microwave and Wireless Technologies: Select Proceedings of ICMWT*, 2021, 543-552. https://doi.org/10.1007/978-981-19-0312-0_53
- [23] Thamizharasan, V. & Kasthuri, N. (2023). FPGA Implementation of Proficient Vedic Multiplier architecture using Hybrid Carry Select Adder. *International Journal of Electronics*, 111(8), 1253-1265. <https://doi.org/10.1080/00207217.2023.2245194>
- [24] Kamrani, H. & Heikalabad, S. R. (2021). Design and implementation of multiplication algorithm in quantum-dot cellular automata with energy dissipation analysis. *The Journal of Supercomputing*, 77, 5779-5805. <https://doi.org/10.1007/s11227-020-03478-6>
- [25] Mohanarangam, K., M Shirur, Y. J., & Choi, J. R. (2023). Accelerating DSP Applications on a 16-Bit Processor: Block RAM Integration and Distributed Arithmetic Approach. *Electronics*, 12(20). <https://doi.org/10.3390/electronics12204236>
- [26] Marimuthu, M., Rajendran, S., Radhakrishnan, R., Rengarajan, K., Khurram, S., Ahmad, S., & Shafiq, M. (2023). Implementation of VLSI on Signal Processing-Based Digital Architecture Using AES Algorithm. *Computers, Materials & Continua*, 74(3). <https://doi.org/10.32604/cmc.2023.033020>
- [27] Aljaedi, A., Rashid, M., Jamal, S. S., Alharbi, A. R., & Alotaibi, M. (2023). An Optimized Flexible Accelerator for Elliptic Curve Point Multiplication over NIST Binary Fields. *Applied Sciences*, 13(19), 10882. <https://doi.org/10.3390/app131910882>
- [28] Varma, A. V. S. S. & Manepalli, K. (2024). VLSI realization of hybrid fast fourier transform using reconfigurable booth multiplier. *International Journal of Information Technology*, 16(7), 4323-4333. <https://doi.org/10.1007/s41870-024-02037-z>
- [29] Mohammed Siyad, B. & Mohan, R. (2024). Processing-in-memory based multilateration localization in wireless sensor networks using memristor crossbar arrays. *Concurrency and Computation: Practice and Experience*, 36(12), e8047. <https://doi.org/10.1002/cpe.8047>
- [30] Ahmad, I., Hussain, T., Shah, B., Hussain, A., Ali, I., & Ali, F. (2024). Accelerated Particle Swarm Optimization Algorithm for Efficient Cluster Head Selection in WSN. *Computers, Materials and Continua*, 79(3), 3585. <https://doi.org/10.32604/cmc.2024.050596>

Contact information:

NIRMAL KUMAR R., Assistant Professor
(Corresponding Author)
Department of Electronics and Communication Engineering,
Bannari Amman Institute of Technology,
Erode, Tamil Nadu, India, 638 401
E-mail: nirmalecebit@gmail.com

VALARMATHI R. S., Professor
Department of Electronics and Communication Engineering,
Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology,
Chennai, Tamil Nadu, India, 600062
E-mail: atrmathy@gmail.com

KALAMANI M., Professor
Department of Electronics and Communication Engineering,
KPR Institute of Engineering and Technology,
Coimbatore, Tamil Nadu, India, 641407
E-mail: kalamani.mece@gmail.com