Neural Processing Letters (2024) 56:125
https://doi.org/10.1007/s11063-024-11495-2

Power Optimization in Wireless Sensor Network Using VLSI Technique on FPGA Platform

Saranya Leelakrishnan[1] · **Arvind Chakrapani**[1]

Accepted: 13 November 2023

Abstract

Nowadays, the demand for high-performance wireless sensor networks (WSN) is increasing, and its power requirement has threatened the survival of WSN. The routing methods cannot optimize power consumption. To improve the power consumption, VLSI based power optimization technology is proposed in this article. Different elements in WSN, such as sensor nodes, modulation schemes, and package data transmission, influence energy usage. Following a WSN power study, it was discovered that lowering the energy usage of sensor networks is critical in WSN. In this manuscript, a power optimization model for wireless sensor networks (POM-WSN) is proposed. The proposed system shows how to build and execute a power-saving strategy for WSNs using a customized collaborative unit with parallel processing capabilities on FPGA (Field Programmable Gate Array) and a smart power component. The customizable cooperation unit focuses on applying specialized hardware to customize Operating System speed and transfer it to a soft intel core. This device decreases the OS (Operating System) central processing unit (CPU) overhead associated with installing processor-based IoT (Internet of Things) devices. The smart power unit controls the soft CPU's clock and physical peripherals, putting them in the right state depending on the hardware requirements of the program (tasks) being executed. Furthermore, by taking the command signal from a collaborative custom unit, it is necessary to adjust the amplitude and current. The efficiency and energy usage of the FPGA-based energy saver approach for sensor nodes are compared to the energy usage of processor-based WSN nodes implementations. Using FPGA programmable architecture, the research seeks to build effective power-saving approaches for WSNs.

Keywords Central processing unit · Field-programmable gate array · Power optimization · VLSI · Wireless sensor networks

✉ Saranya Leelakrishnan
saranyalece@gmail.com

[1] Department of Electronics and Communication Engineering, Karpagam College of Engineering, Coimbatore, Tamil Nadu, India

Published online: 28 March 2024

1 Introduction

In today's society, power usage is a major issue. All portable gadgets, such as laptops, cell phones, are powered by batteries. As a result, the circuits included in gadgets have to be low-power. Size and latency were essential elements throughout the 1970s and 1980s, although there is a trade-off [1]. When a circuit grows more complicated, testing becomes more difficult. Testability must be considered in the design process. However, since 2000, the quantity of portable gadgets with higher power consumption has become a significant concern [2]. Increased power loss causes the temperature to rise too high in a wearable control platform. As the temperature has increased, the incidence of gadget failures has diminished, thereby influencing the system's reliability positively [3]. Reduced power can be achieved at the device layer (complementary metal–oxide–semiconductor (CMOS) technology) or at a higher architectural level. It can make design choices that affect electricity usage while designing a circuit at a greater level [4]. The major quality requirements are the efficiency, attributes, size, and mass of final customers [5, 6]. Low-power solutions are still the way to go if users want to avoid paying extra for things like ventilation and passive heatsinks [7]. Power consumption is a major issue for portable devices, such as laptops and cell phones. As a result, the circuits included in gadgets have to be low-power. However, the increasing number of portable gadgets with higher power consumption is becoming a significant concern. This is because increased power loss causes the temperature to rise too high, which can lead to gadget failures. The motivation for the proposed hybrid power management model is to lessen the power consumption of portable devices, which will extend their battery life, reduce their cost, improve their reliability, and make them more environmentally friendly.

The key contribution of this work is follows,

- The paper proposes a power optimization model for wireless sensor networks (WSNs) called POM-WSN.
- POM-WSN uses a customized collaborative unit with parallel processing capabilities on FPGA and a smart power component to implement a power-saving strategy.
- The customizable collaborative unit reduces the OS CPU overhead associated with processor-based IoT devices by customizing Operating System (OS) speed and transferring it to a soft intel core.
- The smart power unit controls the soft CPU's clock and physical peripherals, putting them in the right state depending on the hardware requirements of the program (tasks) being executed.
- The FPGA is used to implement POM-WSN in hardware, which reduces the entire completion time of real-time tasks, ensuring high task productivity and, reducing task rejection, improving processing.
- The authors evaluate the efficiency and energy usage of the FPGA-based energy saver approach for sensor nodes and compare it to the energy usage of processor-based WSN nodes implementations.

The remaining article is structured as: Sect. 2 discusses the recent related works. The proposed power optimization model for wireless sensor networks (POM-WSN) is described in Sect. 3. Section 4 illustrates the software implementation and analysis. The conclusion and future scope are discussed in Sect. 5.

2 Related Works

Among the numerous studies related to energy efficient schemes in WSN, some of the most recent research works are reviewed in this section,

Shafiq et al. [8] presented on energy-efficient routing schemes in wireless sensor networks (WSNs). Idle periods occur in WSN for various reasons, including process-required time-varying monitoring of different sensors. Good efficiency is predicted to meet the industry need in some WSN applications.

Kaur et al. [9] have presented the recent advances in MAC protocols for the energy harvesting based WSN. As a result, lower utilization WSN designs are encouraged for long-term application use. As a result, reduced energy consumption contributes to cheaper operating costs and a smaller battery capacity and a reduction in the size of the batteries.

Xu et al. [10] have presented an energy-efficient region source routing protocol for lifetime maximization in WSN. To develop energy WSN, it is necessary to examine the heat dissipation feature of WSN. Energy consumption in WSN is caused by several variables, including network protocols energy usage and sensor network level energy usage. In WSN, power-efficient data transmission is regarded as a unique problem.

Maheshwari et al. [11] have presented Energy efficient cluster based routing protocol for WSN using butterfly optimization algorithm and ant colony optimization. Nevertheless, the time to accomplish a project in a WSN's information computer nodes is substantially greater than the time it takes to send data in a sending node.

Patil et al. [12] have established the cookies, Hire Cookies, and other platforms. This module is ideal for applications that require quick processing and a sophisticated sensor interface. The FPGA handles activities linked to complicated sensor connections, while the microcontroller handles communication.

Chéour et al. [13] investigated the trends of FPGA used for low-power wireless sensor network to conserve energy, and renewable energy is included in the runtime smart power architecture. Sustainable energy is a viable way for prolongs the length of battery-powered portable systems by overcoming the power supply restriction.

Patil and Deshpande [14] introduced a novel capacity constraint-aware approach that reduces unreliability due to temperature-aware techniques' low efficiency and instability. For the same reason, scholars created a paradigm to sustain performance. FPGA technology is gaining popularity in integrated devices.

Toubal et al. [15] have described the FPGA implementation of a wireless sensor node with built-in security coprocessors for secured key exchange and data transfer. It investigated the system for digital logic, such as fine-grained system. FPGA power saving has become critical in the development of trustworthy embedded systems.

Misra et al. [16] have been introduced FPGA platforms. This method is used for computationally expensive memory processes. Each VF domain comprises probes that identify idle time owing to memory access issues or congestion.

Mishra and Kumar [17] have been presented architecture for dedicated hardware. The technique was adopted on the FPGA platform by using semiconductor regulators, which reduces the latency produced by off-chip authorities. Similarly, scholars utilized Adjustable Voltage Supplying (look-ahead technique) for power savings in motion estimation processors.

3 Proposed Power Optimization Model for Wireless Sensor Networks (POM-WSN)

The processing element, sensor, wireless transmitter, and power supply are the basic elements of a sensor network. Its power supply (battery) has a finite lifespan. To satisfy application requirements, sensor system components must have a longer lifetime. The WSN's processing element is a crucial component. It's used to operate sensors, acquire information from sensors, display the data obtained, and run protocols. This unit receives the transmitted data from the ADC (Analog to Digital Converter) for management and processing. In addition to the components that make up a WSN network, the processing element is responsible for managing the sensing node's ADC activity. It establishes a connection with other Network elements.

The hardware architecture of the POM-WSN system is depicted in Fig. 1. The hardware components such as a transceiver, ADC (Analog to Digital Converter), microcontroller, power source and sensors are used. An external memory card is used to store the data collected from the WSN. The processor unit's power usage characteristics necessitate a strict power-saving method. The processing element chosen is critical since it must achieve the specified requirements of the WSN program while consuming minimal power. Energy optimization is a key challenge in wireless sensor networks (WSNs) due to the limited battery life of sensor nodes. The proposed POM-WSN model is a flexible and powerful tool that can be used to implement a wide range of energy optimization techniques in WSNs. For example, POM-WSN can be used to cluster sensor nodes, aggregate data, put sensor nodes into power saving modes, and adjust clock speed and voltage dynamically. POM-WSN has the potential to significantly extend the battery life of sensor nodes and enable them to operate for longer periods of time.

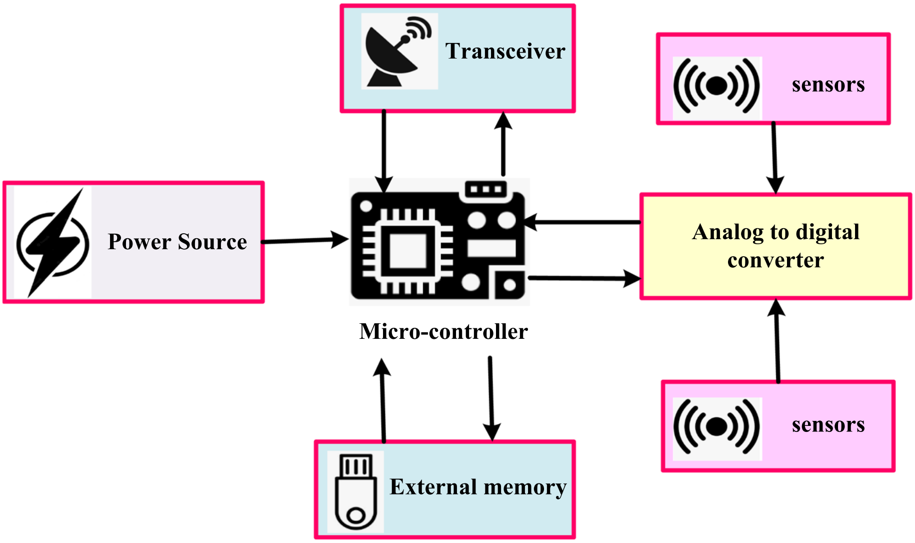


Fig. 1 Hardware architecture of the POM-WSN system

3.1 Working of Hardware Architecture of the POM-WSN System

This sensor network unit can be an active or passive electrical fibre. The amount of processes in communications, such as modulation, filtration, decoder, and multiplexer, increases the complexity and cost of transceiver operation. The transponder is used to communicate wirelessly with neighboring nodes and the outside world. Several components influence energy usage, such as modulation type, data transfer rate, telecommunications power, and operating repetition rate. Data-intensive apps create a greater volume of data that must be delivered via the Internet. Choosing the right cordless transceiver for dependable data transfer and low power consumption is critical. The scale of the system, the region to be served, and extra overheads all influence the transceiver choice. The pace of transmitting data determines the energy usage of the transmitter. RF communication is favored in WSNs where the number of packages to be delivered shortly and the transmitting data rates are very low. Spectrum reuse is achievable due to the small transmission distance. Transceiver information transmission with a low duty ratio is desirable to save electricity.

The detector is a sensor or component that detects and converts physical variables into electrical signals. Active and passive detectors are the two categories of sensors. Active detectors constantly monitor the environment, necessitating a constant supply of electricity from the energy source. Passive detectors are self-powered and detect information without the need for dynamic probes to deal with changing environmental circumstances. Signal filtering, translation of a measurable signal to an electronic current, and analog to digital translation are all examples of power usage in sensors. The analog sensor information is converted to digital to use an ADC (Analog to Digital Converter) and then transferred to the processing system for analysis and control. As a result, the sensor is a component of WSN node that is accountable for energy usage.

For operating system (OS) speed, four distinct techniques have been suggested:

- Operating system as a coprocessor: The operating scheme is installed as a coprocessor. As a result, OS activities are routed to the coprocessor, and the coprocessor handles all OS tasks.
- Characteristics of the operating system in equipment as an element: The sophisticated and full mechanism for OS operations is usually implemented to decrease OS overhead.
- Hardware OS: The complete OS is represented as a physical object.
- OS as customized instructions: Task planning and timer handling are common functions in customized commands.

Modifying hardware platforms and applications to the soft-core central processing unit (CPU) accelerates the operating system. The OS CPU inefficiency is reduced in the proposed concept by using a modified instructions language and physical component to accelerate the OS. As a result of the reduced OS CPU latency, more working time is available for a user program. The commonly conducted OS activities are discovered and translated into customized instruction (CI) or hardware components as part of the treatment. As a result, a sequence of instructions was reduced to fewer processes, lowering the OS CPU burden. The execution of user tasks is given plenty of time. When a customized instruction is performed all instruction portions run in parallel. The custom command is examined either using higher-level or assembly-language programming.

The theoretical flow of growth involves the following steps:

Step 1: Create a soft intel core on an FPGA and programme it with a real-time operating system (RTOS).

Step 2: Create a customized cooperative unit and put it into action.

- Determine which RTOS procedures are used frequently (task planning, timer tick control, interrupt control, and more features are available.)
- Using FPGA, design and build these elements in two ways: as an element; following specific requirements
- Use a soft core processor to implement them.
- Examine and compare the findings, considering various factors such as FPGA area, efficiency, and energy usage.

Step 3: Plan and build a power control unit (clock gating hybrid approach).

- Examine the collaborative customized unit's servo controller.
- If the needed speed is less or more, turn on the component. The soft CPU's voltage and current are adjusted according to performance requirements.
- Start the clock tree unit if the CPU is idle. It turns off the processor's and accessories' clocks in leisure time.

Step 4: Examine the soft core CPU's power usage.

Step 5: Evaluate the power usage findings of the CPU without using this approach to the outcomes of the suggested system.

3.2 Process of Proposed Hybrid Power Management Scheme (POM-WSN)

This scheme aims to deliver power savings via time gating in response to the control system provided by the cooperating unit. The timer is accountable for action in the circuitry, and power loss is mostly dependent on signal transitions in activities in the loop. The clock triggers all portions of the circuit. If any portion of the circuitry is presently not functioning, its timer is deactivated, resulting in less signal shifting and lower power loss for that section. As a result, clock gating is seen as a very straightforward strategy for reducing power consumption. Another power-saving technology scales the voltage and frequency to meet the needs. The power wasted in a sensor network is split into dynamic and static energy loss. The actual power is denoted P_{ac}, and expressed in Eq. (1).

$$P_{ac} = P_{st} + P_{dy} \tag{1}$$

The static power and dynamic power are expressed as P_{st} and P_{dy}. The actual power is again represented in Eq. (2).

$$P_{ac} = C_{lea} \times v + K_{eff} \times v^2 \times fr \tag{2}$$

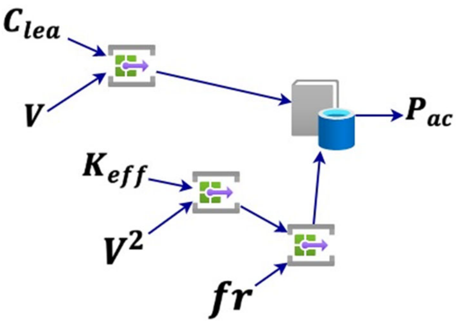
where K_{eff} is denoted as CMOS load capacitance, V is expressed as Supply Voltage, C_{lea} is denoted Leakage current, and fr is denoted Operating Frequency.

Figure 2 represents the graphical view of mathematical operation P_{ac}. The leakage current is denoted as C_{lea}, K_{eff} is denoted as the load capacitance, and frequency of the operation is denoted fr. This module aims to use a hybrid engine strategic planto reduce energy usage while meeting a performance requirement. Clock gating reduces static power, whereas the approach reduces dynamic power. The customized collaborative unit is an intelligent portion of the sensor network that saves CPU overhead. It also maintains the smart power unit and provides control signals to engage the clock tree modules as required. The following are the stages involved in implementing the prototype:

Step 1: Create a voltage/frequency scalability model for the processor.

The following criteria are taken into account for the scaling prototype. $Ex : E_1, E_2, E_3$ Methods were created as special instructions for the task given. Range of frequencies is

Fig. 2 The graphical view of mathematical operation P_{ac}



expressed as $fr_{\min} - fr_x - fr_{\max}$. Range of voltages is expressed as $vl_{\min} - vl_x - vl_{\max}$. Scaling paradigm for processors $fr_{\min}, \alpha, \Delta fr, fr_{\max}$ is expressed in Eq. (3),

$$\alpha = \alpha_1, \alpha_2, \alpha_3, \ldots\ldots..\alpha_m \tag{3}$$

where Δf is the point at which the frequencies can be changed and the scalability is expressed as α_m.

Step 2: Calculate the program's workload to be run on the scheme: Workload can be calculated using a command signal from a bespoke cooperating unit.

Step 3: Establish the power usage concept: This power usage model is optimized by taking the deadlines (Dl_x) into account. It assists in determining the desired frequency (fr_x) and voltage (v_x). The power usage is expressed in Eq. (4).

$$P = \sum_{x=0}^{M} K_x \frac{\rho_x}{b_x} (v_x)^2 \tag{4}$$

where K_x is the task's entire capacitance and ρ_x is the task's shifting activity. The biasing condition is expressed as b_x. The supply voltage is expressed as v_x. A CPU's clock speed is proportionate to its power supply, and it is expressed in Eq. (5).

$$v_x = C(fr)^2 \tag{5}$$

when C is the CPU's standard and the frequency is denoted as fr. The power is computed in Eq. (6)

$$P = \sum_{x=0}^{M} K_x \frac{\rho_x}{b_x} \times C(fr_x)^2 \tag{6}$$

The clock signal fr_x is adjusted to meet the task's deadlines and other limitations. K_x is the task's entire capacitance and ρ_x is the task's shifting activity. C is the CPU's computation time. The biasing condition is expressed as b_x. As a result, the equivalent power usage is updated using Eq. (7).

$$\Delta P = \sum_{x=0}^{M} K_x \frac{\rho_x}{b_x} \times C\left[(fr_x)^2 \times \frac{(\alpha_x)^2}{\rho_x \times fr_{\min} \times \Delta fr}\right] \tag{7}$$

The above code is used to find the best-scaled coefficient (α_x) for each job E_x power usage (ΔP) is minimized while goals are met. The biasing condition is expressed as b_x. The minimum frequency is expressed as $fr_{\min}$. K_x Is the task's entire capacitance and ρ_x is the task's shifting activity. The deadline restriction is expressed in Eq. (8).

$$\frac{b_x}{fr_{\min} + (\rho_x \Delta fr)} \leq Dl_x \tag{8}$$

The deadline is expressed as Dl_x, the frequency deviation is denoted as Δfr, and the minimum frequency is expressed as $fr_{\min}$. The scaling procedure is as follows if the necessary clock speed is better than the normal frequency:

- Lower the voltage and replace it with the new value (v_x).
- Until the power falls to an equal value, the system runs at the same clock.
- Set the current clock rate (fr_x) by lowering the frequency.
- The scaling procedure is as follows if the desired clock speed is less than the existing frequency.
- Set the current clock rate (fr_x) and increase the speed.
- Set the current–voltage (v_x) by increasing the voltage.
- The system continues to run at the current system clock till the voltage has stabilized at the value equation.
- Clock administration is used to enable and disable associated peripheral clocks selectively. The peripherals necessary for present operation are recognized, and then the clock tree approach is used to start and stop them selectively.

3.3 Proposed Methodology Evaluation Method

This system is constructed on the Altera, and tests are done. The power management flow of the POM-WSN system is shown in Fig. 3. The soft processor is used to analyse the real-time operating systems. The real-time operations of WSN are accessed and then designed.

The implementation takes place before the power analysis stage. The power management unit monitors and controls the power usage in WSN using the FPGA model. The following are the primary phases for deployment that have been proposed:

(1) Set up the processor
(2) Create a customized collaborative unit and put it into action. The methods below are investigated to measure the reduced power usage and increase in performance.

- Operation using simply a software operating system
- Customization of the operating system for implementation
- Compare the outcomes by considering runtime, energy usage, and logic aspects.

(1) Create a power control system and put it in place.
(2) Examining the overall outcome

The following scenarios are used to compute performance factors such as slice area, energy usage, and processor processing time. No two-time gating techniques are employed in the customized collaborative unit mixed with FPGA softcore CPU. Clock gated approaches are used with a collaborative custom device and an FPGA soft intel core. Quartus III includes the Time Quest timing analysis and the Power Play energy analysis tools. The power usage and processing time may be monitored using these techniques.

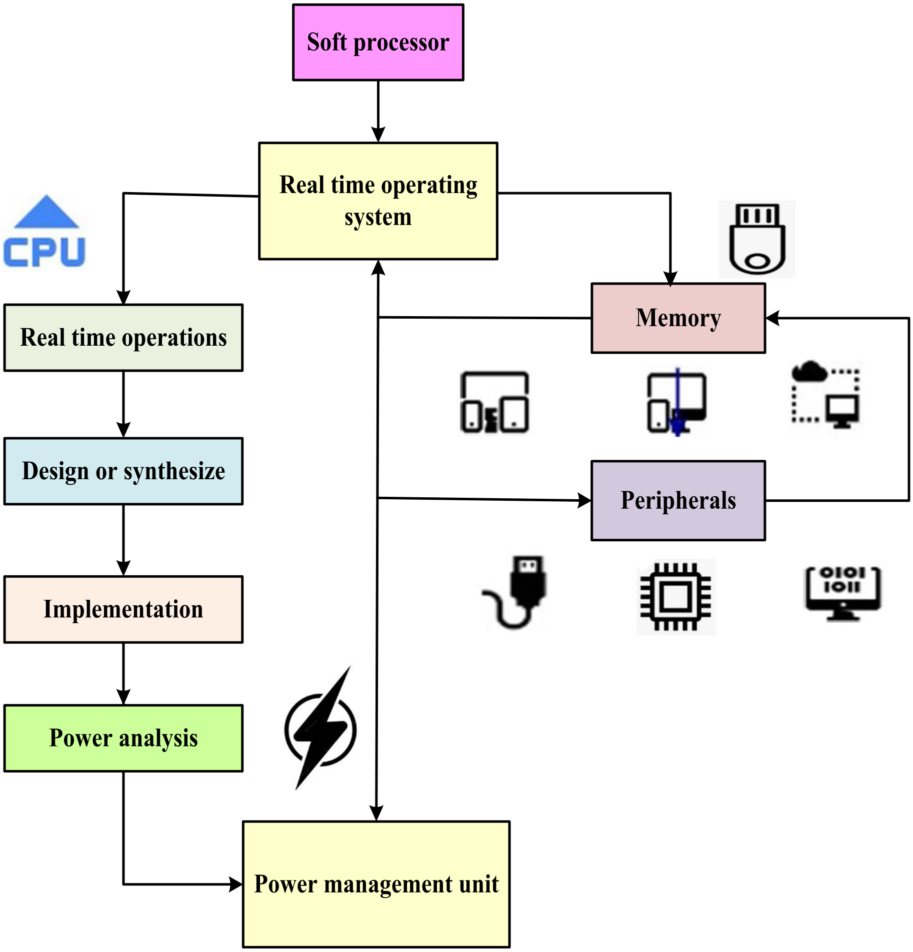


Fig. 3 The power management flow of the POM-WSN system

3.4 Power Estimation Methods

In recent years, one of the main issues for FPGA providers has been power estimation. Designers regulate the efficiency of a range of implementations by using efficient and precise power estimating techniques. The switching activity determines how much power is dissipated, and therefore the most active circuit consumes the most power. The power wasted during every transition's ascending and descending periods due to the passage of current straight from the earth is referred to as short circuit energy dissipation. Setting it to 10% energy loss is the simplest method to evaluate energy loss due to a short connection. As a result, it solely evaluates the mean voltage levels of power estimate approaches. The dynamic power is computed in Eq. (9),

$$P_{dy} = \frac{1}{2}(V_{dd})^2 \left(\frac{fr_{clk}}{\sum_{y=0}^{M} K_y \rho_y} \right) \tag{9}$$

where fr_{clk} denotes the clock signal, V_{dd} denotes the voltage, and K_y, ρ_y. Accordingly, the load impedance and the amount of changes per clock pulse of the y^{th} node. The above formula is used to calculate the overall number of changes for each circuit terminal, which is used to estimate power. In regular Shifting, the power estimate is fairly straightforward. If a node has K_y reaction capacitor and a local oscillator with frequency (fr) is created, the power wasted on the median is $(V_{dd})^2 fr_{clk} K_y$, where V_{dd} is the potential that is controlled. The input sequence usually determines the shifting activity, and this system-building power is tricky.

The pictorial representation of the function P_{dy} is shown in Fig. 4. It uses the clock frequency, the adaptation ratio and coefficient functions are used to compute the results. It is nearly impossible to mimic the circuits of all conceivable inputs for these complicated systems. The major inputs should be kept distinct. The key reason for requiring personal power is to locate the circuit elements that utilize the most power. In a particular iteration, the proportion of phases in which changes and the power level of an original signal is estimated using probabilistic approaches.

Altera provides FPGA architects with the Power-play Power Analyst system, which allows them to calculate power once the design is completed. Altera also provides the Quartus II Initial Burst Estimation, an initial burst estimator file. The user normally includes the implementation to execute using any specialized experimental results or testing tools to

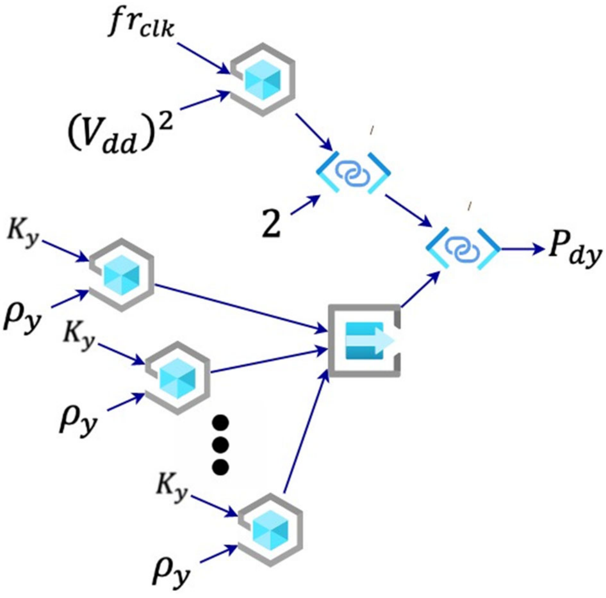


Fig. 4 The pictorial representation of the function P_{dy}

assess the CAD result. Power-play is an energy analyzer that calculates power quality and matches the target machine to determine how the output architecture is handled and placed. The Power-play energy analyzer can engage four different signal streams. The total wasted capacity may be separated into two sections: one owing to IO and VCC port currents, and the other due to a 3.3 V power source. The energy analyzer Power-play assesses power quality and modifies the objective for understanding the layout installed and routed once the design is synthesized. By putting the ammeter on the instrument attached to this resistance, the ammeter may be accurately observed.

3.5 Power-Play Energy Analyzer

In this section analyze the power optimization techniques that involve multiplication of matrices and hardware design that are discussed below:

3.5.1 Multiplication of Matrices

It examined and compared efficiency and electricity utilization in three distinct designs by multiplying the A and B matrix of n to 32-bit elements and creating an n to 64-bit C matrix. The first models use FPGA Gadget with varying $2n$ and $2n_2$ entries for memory, while the third approach uses the NIOS IIe architecture. Here, n represents the number of rows and columns in matrices A and B and n_2 represents the number of rows and columns in matrix C. In contrast to the multipliers in FPGA, it has proved that NIOS IIe is not even an option for the renewable energy of multiplication $n \times n$ matrix. It depicted a fairly standard method, design, and resources employed in three completely different matrix implementations. Our solutions fall into two design categories:

- Architecture based on General Purpose Processors
- Architecture based on a Single Purpose Processing

A solution for Altera NIOS architecture multiplies the array using a simple and basic conversion in the chip memory. The very first VHDL program is meant in hardware, while the second VHDL system is developed in software; neither solution employs a new application store to multiple the matrix, but rather many different types of equipment. Physically first model VHDL obtains register for storage devices, m multipliers, and m-1 continuous adders. In contrast, physical second conceptual VHDL gets entries for temporarily held components, m multipliers, and m-1 cascaded full adder. It employed a computer system built on a basic core CPU that employs user-friendly multiplying real outcomes with a memory and a summing convention is shown in algorithm 1.

```
$for(i = 0; i < n; i++)\{$
$for(j = 0; j < n; j++)\{$
$for(k = 0; k < n; k++)\{$
$K[i][j] = K[i][j] + \frac{B[i][k]}{D[K][j]}$
        end for
    end for
end for
```

In NIOS IIe, there is no HW multiplier, and NIOS II employs the Shifting Add multiplying method, which has a far higher throughput than HW multiplying methods. For the first equipment problem, matrix multiplying employs temporary 2 m registers to contain elements with m-1 cascaded full adder and m multiplication. Different memory regions with matrices A, B, and C are employed in design structures. So the first columns of the B grid are nearly always read in combination with any matrices of the A row; it needs A readings and the B matrices $(n - 1)n$ readings to calculate lines in the C matrix. The delay for m lines is calculated in Eq. (10)

$$I = \left[j + \frac{j-1}{j} \right] \times j \tag{10}$$

The input features of WSN are denoted as j. Because the shared memory gap stays at three, the first m registries are allocated to rows A, while the following m logs are allocated to columns B. To compute the full C matrix, every line in A must be discovered just once, whereas the B columns must be discovered m times. By eliminating the production of the multiplication, this parallel response time only through a formula of lag and enables matrix multiplication to be detached by system memory, allowing computed components to be authored correspondingly and the first section of the B lattice to be perused in comparison to any queue of matrix A. The diagram of time wherein the matrix C elements are accessible in the adder reply and at C block data blocks.

3.5.2 Hardware Design 2—VHDL Program

It kept m-1 cascaded full adder and m multiplication with the same design even with a prior approach in the Grid solution. However, it can utilize 2 × 2 to save space full-filled gaps if it uses storage locations registries instead of 2 m registries. The 2 × 2 registries and the procedure above have decreased the scale of such difficulty to two. In the first m2 rounds, it keeps A-lines and B columns while computing the initial C lines since the data is available. During this time, storage occurs after each n cycle of an A matrix liner, a B matrix column, and a one-off in the C row commencing to be computed. Following m2 clock strikes, B and A terms are pre-placed, and no external memory usage is required to obtain B and A words; at this point, leftover C matrix concepts are formed at every clock pulse. Delay is expressed in Eq. (11).

$$I = 2j^3 - \frac{j^2}{n+1} + j \tag{11}$$

The time at the adder's reply and the storage of block C is depicted. The WSN feature is denoted as j. The wireless sensor network-based power optimization model is designed using the VLSI design and FPGA model. The software analysis of the POM-WSN system is analyzed and evaluated in the next section.

Computational Complexity The time complexity when fitness estimation with maximum iterations,$Maxgen$, count of objectives,f and population size, $Popsize$ implies $O(Maxgen \times f \times popsize)$. The time taken to initialize the populace implies $O(popsize \times f)$. The proposed Power Optimization requires $O(f \times (s + popsize))$ time for updating the archive in non-dominated solutions, where s denotes optimal search agents. Thus, the entire time complexity of the proposed algorithm is $O(Maxgen \times f \times (popsize + s) \times L)$ and space complexity during generation of population in memory requires $O(popsize \times f)$ time. The process of fitness computation and the updation of archive repeats until the algorithm attain maximal count of iterations. The time complexity of Power Optimization with N input denotes $O\left(N^3\right)$. The whole time complexity of the proposed Power Optimization in Wireless Sensor Network using VLSI Technique on FPGA Platform is $O\left(N^3 \times Maxgen \times f \times popsize\right)$ respectively.

4 Software Analysis and Evaluations

In this work, the implementation is done using the Terasic de5-net toolkit from Intel (Altera) supports the Stratix V FPGA processor. It ran the tests on this Intel FPGA board, which has the following specifications: Altera Stratix FPGA, 60 MHz Oscillator, SDRAM, and 4 parallel communication interface (PCI) Express solid IP blocks with PCI Gen1/2/3 capability. The FPGA chip's Rectangle Regions (RRs) are also utilized to conduct functions on ground station digital to analog converter (DAC) and servers.

The VLSI design using the FPGA design procedures helps to simplify the wireless sensor network design and thus reduces the static and dynamic power consumption. The lower power consumption leads to the highest effectiveness of the POM-WSN system. The software evaluation of POM-WSN system is discussed in this segment. The simulation outcomes like accuracy, efficiency, dissipated power, power error estimation, dynamic power and static power are analyzed for the POM-WSN system and the results are compared with the existing models. The POM-WSN system shows higher simulation outcomes with the help of WSN and FPGA models.

The power estimation error analysis and dissipated energy analysis of the POM-WSN system are represented in Figs. 5 and 6. The FPGA system is designed using VLSI system namely 2 × 2, 4 × 4, 8 × 8, 16 × 16, 32 × 32 and 64 × 64 respectively. The error at the estimated power, and dissipated energy are computed, and the results are plotted. As the design size increases, the energy consumption also increases. The POM-WSN system with wireless sensor network design, and power optimization model ensures the higher efficiency of the proposed system.

Table 1 tabulates the simulation parameters in the Riverbed Modeler. The nodes in 100*100 m area generated randomly. The system implemented simulations over 5, 10, 25, and 50 nodes with diverse payloads for as a minimum four iterations.

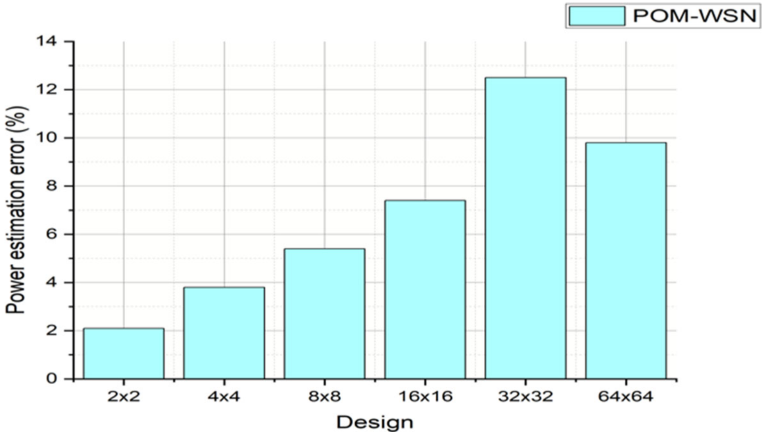


Fig. 5 Power estimation error analysis of the proposed POM-WSN system

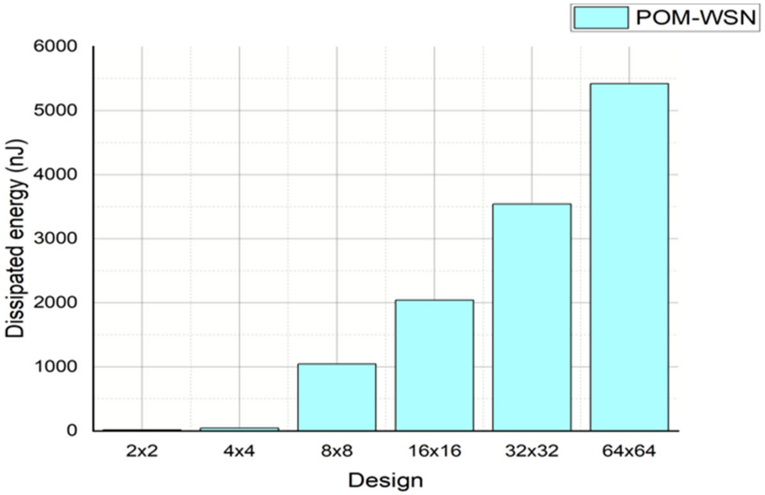


Fig. 6 Dissipated energy analysis of the proposed POM-WSN system

4.1 Comparative Analysis of Evaluation Metrics Based on FPGA Dessign

The POM-WSN system is designed using the VLSI design concepts under FPGA and wireless sensor networks. The simulation outcomes of POM-WSN system is compared with the existing models such as support vector machine (SVM) [18], deep neural network (DNN) [19], fuzzy logic (FL) [20], principal component analysis (PCA) [21], and linear discriminant analysis (LDA) [22] respectively. The POM-WSN system exhibits lower static and dynamic power consumption than the existing models.

Figure 7 depicts static power consumption analysis. The proposed POM-WSN system has attained 25.67, 18.96, 28.35, 16.90, and 21.34% lower static power consumption than the existing methods like SVM, DNN, FL, PCA, and LDA respectively.

Table 1 Simulation parameter table

Parameter	Value
Amount of nodes	3, 5, 10, 25, 50
Kinds of sensors	MICAz
Sensor transmission range	100 feet
Data rate	256 kbps
Payloads	16, 32, 64, 128, 256, 512, 1024 Bytes
Nodes frequency	2.4 GHz
Initial energy	2 AA (1.5 V, 1600mAh)
Area size	100*100 m
Simulation period	30 min
Seed	128
Path loss model	Free space

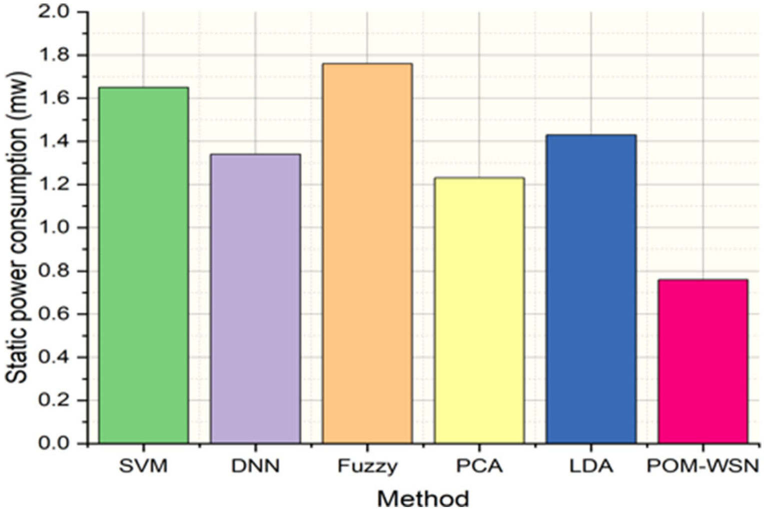


Fig. 7 Static power consumption analysis of the POM-WSN system

Figure 8 shows dynamic power consumption analysis. In this, the proposed POM-WSN system has attained 14.76, 26.87, 16.49, 23.97, and 21.06% lower dynamic power consumption than the existing methods, like SVM, DNN, FL, PCA, and LDA respectively.

Figure 9 displays Network Lifetime analysis. The proposed POM-WSN system has attained 14.76, 26.87, 16.49, 23.97, and 21.06% lower dynamic power consumption than the existing methods, like SVM, DNN, FL, PCA, and LDA respectively.

Figure 10 shows Residual Energy analysis. The proposed POM-WSN system has attained 14.76, 26.87, 16.49, 23.97, and 21.06% lower dynamic power consumption than the existing methods, like SVM, DNN, FL, PCA, and LDA respectively.

Figure 11 shows the efficiency analysis of proposed POM-WSN approaches with PCA model. The POM-WSN system is evaluated under different iterations varying from 5 to 60 with step count of 5 iterations. When the iteration count raises, the respective simulation

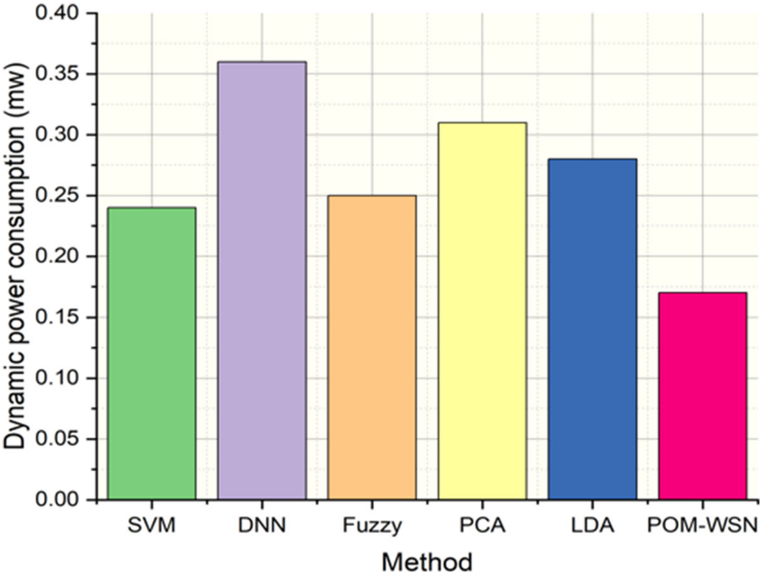


Fig. 8 Dynamic power consumption of the POM-WSN system

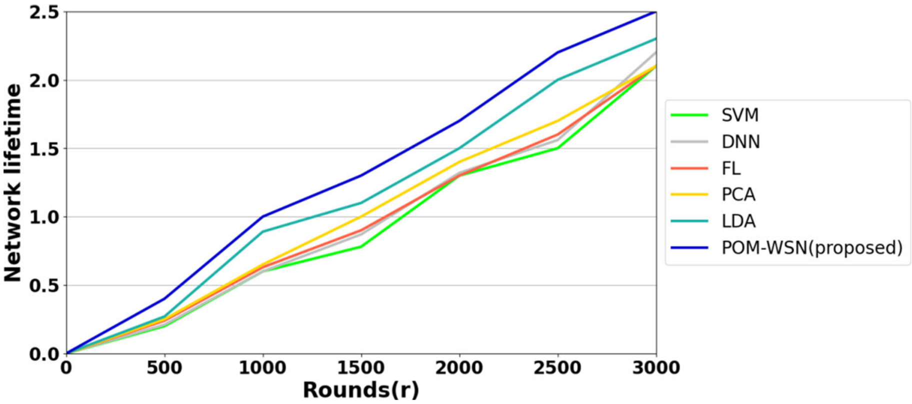


Fig. 9 Network Lifetime of the POM-WSN system

outcome, such as efficiency of the POM-WSN system also increases. Here, the proposed POM-WSN system has attained higher efficiency than the existing PCA.

Figure 12 shows the accuracy evaluation of the POM-WSN system. The POM-WSN system is evaluated under different iterations varying from 5 to 60 with step count of 5 iterations. When the iteration count raises, the respective simulation outcome such as the accuracy of the POM-WSN system also increases. In this, the proposed POM-WSN system has attained higher accuracy than the existing PCA.

Figure 13 displays the comparison of computational complexity. Here the proposed POM-WSN methods CPU operation time and memory usage are linearly increased. Here the time

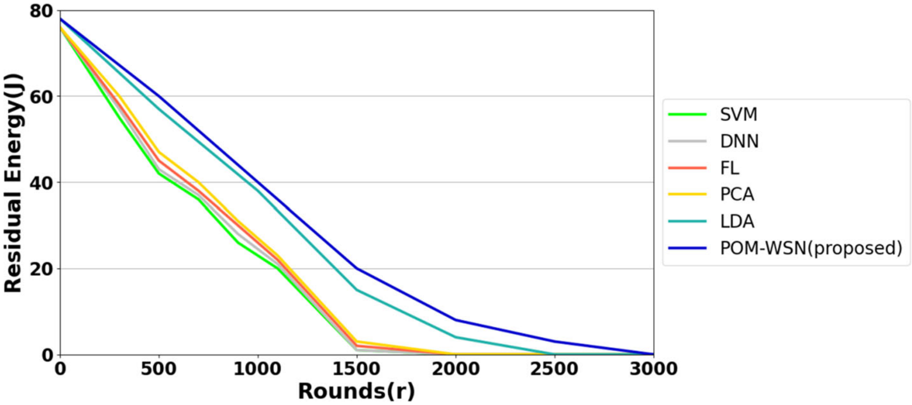


Fig. 10 Residual Energy of the POM-WSN system

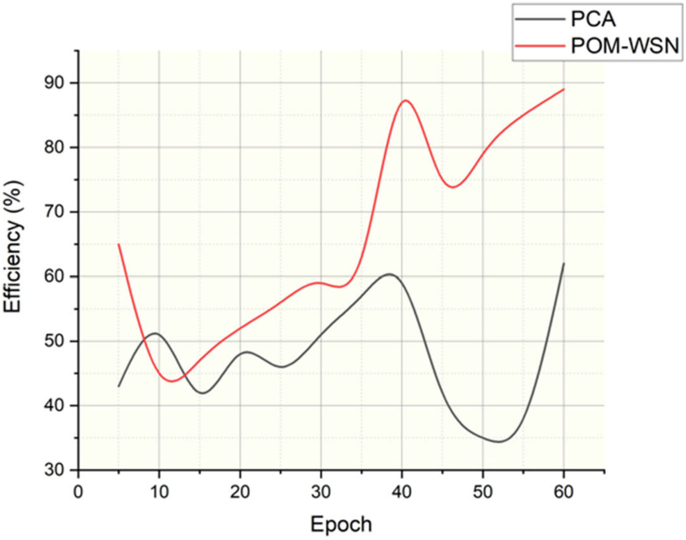


Fig. 11 Efficiency analysis of the POM-WSN system

complexity of the proposed POM-WSN method shows lower than the existing methods respectively.

4.2 Comparative Analysis of Evaluation Metrics Based on Non-FPGA Models

The simulation outcomes of POM-WSN system is compared with the existing non-FPGA models like ARM Cortex-M7 processor [23], Intel Pentium-class [24], Intel Pentium x86processor [25], IBM Bluemix cloud service [26], and Intel Pentium 4 Prescott [27] respectively.

Table 2 indicates the static along dynamic power consumption analysis of POM-WSN

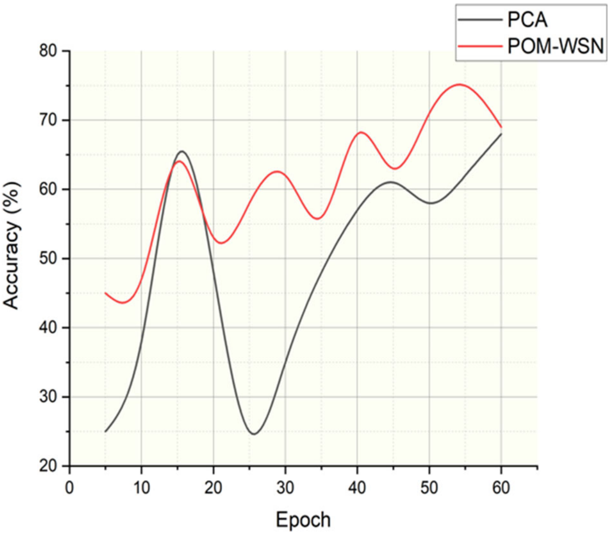


Fig. 12 Accuracy analysis of the POM-WSN system

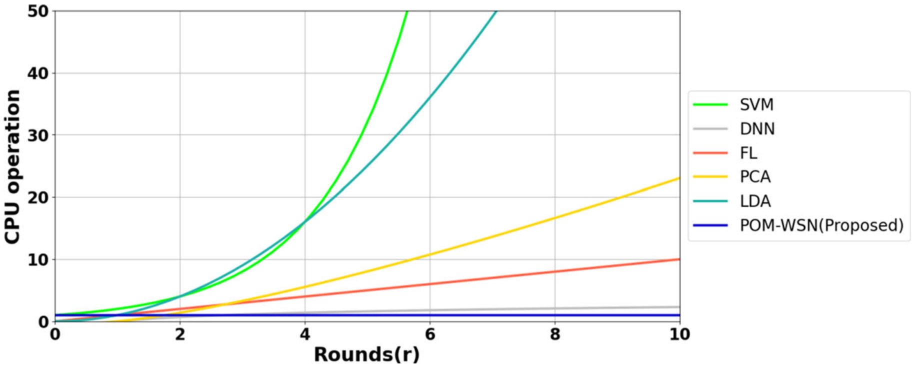


Fig. 13 Computational complexity

Table 2 Power consumption analysis

Model	Static power consumption (mw)	Dynamic power consumption (mw)
ARM Cortex-M7 processor	2.74	1.32
Intel Pentium-class	2.15	1.96
Intel Pentium × 86 processor	1.96	1.67
IBM Bluemix cloud service	1.75	1.73
Intel Pentium4 Prescott	2.64	1.52
POM-WSN (Proposed)	1.24	1.25

Table 3 Power analysis of the POM-WSN system

Model	Power estimation error (%)	Dissipated energy (nJ)
ARM Cortex-M7 processor	4.2	874
Intel Pentium-class	3.9	1078
Intel Pentium × 86 processor	6.3	2431
IBM Bluemix cloud service	8.3	1748
Intel Pentium4 Prescott	7.6	985
POM-WSN (Proposed)	2.1	532

system. The POM-WSN system is designed with the VLSI system and FPGA design concepts. The POM-WSN system utilizes lower static and dynamic power consumption than the non-FPGA models such as ARM Cortex-M7 processor [23], Intel Pentium-class [24], Intel Pentium × 86 processor [25], IBM Bluemix cloud service [26], and Intel Pentium 4 Prescott [27] respectively. The hybrid power optimization model reduces the total energy consumption in the VLSI system. As the wireless sensor network nodes in the environment increase, the respective power consumption also increases. But the POM-WSN system with a hybrid model utilizes minimal power and produces higher efficiency.

Table 3 tabulates the power analysis of the POM-WSN system. The power estimation error at the output and the dissipated energy are computed with respect to the design. The proposed POM-WSN results are compared with ARM Cortex-M7 processor, Intel Pentium-class, Intel Pentium × 86 processor, IBM Bluemix cloud service, and Intel Pentium4 Prescott respectively. The wireless sensor network module is designed using the VLSI design utilizing the FPGA design. As the design complexity increases, the error and the dissipated energy at the output also increase. The power optimization model is used to compute the power utilized in the FPGA system and reduces the static and dynamic power consumption. The simulation outcomes show the effectiveness of the POM-WSN system than the other designs.

4.3 Discussion

The article's focus on addressing power consumption challenges in wireless sensor networks (WSNs) is both timely and significant. With the growing demand for high-performance WSNs, power requirements have become a critical concern, endangering their sustainability. The article introduces a novel approach, the Power Optimization Model for Wireless Sensor Networks (POM-WSN), which leverages VLSI-based power optimization technology. The study recognizes that various elements within WSNs, including sensor nodes, modulation schemes, and data transmission, impact energy usage significantly. By proposing a system that combines FPGA-based parallel processing capabilities with a smart power component, the article offers an innovative solution to reduce energy consumption effectively. The customizable collaboration unit, which optimizes Operating System functions, and the smart power unit, which manages clock speeds and peripherals based on program requirements, collectively promise a substantial reduction in energy overhead compared to conventional processor-based IoT devices. The article's commitment to benchmarking the FPGA-based energy-saving approach against traditional processor-based implementations demonstrates its practical approach to enhancing WSN power efficiency. Overall, this research presents a promising avenue for addressing the pressing power consumption challenges in WSNs, with potential implications for the broader Internet of Things (IoT) ecosystem.

5 Conclusion

The design layout of a power-saving strategy for FPGA softer core-based sensing nodes is implemented in this study. Conventional power-saving strategies used in devices lack the essential flexibility, stability, and effectiveness. A power optimization model for wireless sensor networks (POM-WSN) is proposed in this article. This proposed solution reduces the OS-related CPU overhead, also managing power consumption. Experimentation is done by developing a system that includes a customized cooperative unit, a power control unit, and a softened core CPU, then simulating and executing it on the FPGA board. These simulation and execution responses would be further evaluated and compared to the benchmarking methodologies.

Acknowledgements Not applicable

Author Contribution Mrs. Saranya Leelakrishnan-(Corresponding Author)—Conceptualization Methodology, Original draft preparation. Mr. Arvind Chakrapani-Supervision.

Funding This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Data Availability Data sharing does not apply to this article as no new data has been created or analyzed in this study.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethical Approval and Consent to Participate This article does not contain any studies with human participants performed by any of the authors.

Consent for Publication Not Applicable.

Human and Animal Rights Not Applicable.

References

1. Kochhar A, Kaur P, Singh P, Sohi BS (2020) MLMAC-HEAP: A multi-layer MAC protocol for wireless sensor networks powered by ambient energy harvesting. Wireless Pers Commun 110:893–911
2. Jeske M, Rosset V, Nascimento MC (2020) Determining the trade-offs between data delivery and energy consumption in large-scale WSNs by multi-objective evolutionary optimization. Computer Network 179:107347
3. Haseeb K, Islam N, Almogren A, Din IU (2019) Intrusion prevention framework for secure routing in WSN-based mobile Internet of Things. IEEE Access 7:185496–185505
4. Feng S, Shi H, Huang L, Shen S, Yu S, Peng H, Wu C (2021) Unknown hostile environment-oriented autonomous WSN deployment using a mobile robot. J Netw Comput Appl 182:103053
5. Ramshankar N, Joe Prathap PM (2023) Reviewer reliability and XGboost whale optimized sentiment analysis for online product recommendation. J Intell Fuzzy Syst 44:1547–1562
6. Panahi U, Bayılmış C (2023) Enabling secure data transmission for wireless sensor networks based IoT applications. Ain Shams Eng J 14(2):101866
7. Hou L, Chen W (2020) A novel MPPT method for autonomous wireless sensor networks node with thermal energy harvesting. Eng Res Express 2(1):015005
8. Shafiq M, Ashraf H, Ullah A, Tahira S (2020) Systematic literature review on energy efficient routing schemes in WSN–a survey. Mobile Netw Appl 25:882–895
9. Kaur P, Sohi BS, Singh P (2019) Recent advances in MAC protocols for the energy harvesting based WSN: a comprehensive review. Wireless Pers Commun 104:423–440
10. Xu C, Xiong Z, Zhao G, Yu S (2019) An energy-efficient region source routing protocol for lifetime maximization in WSN. IEEE Access 7:135277–135289
11. Maheshwari P, Sharma AK, Verma K (2021) Energy efficient cluster based routing protocol for WSN using butterfly optimization algorithm and ant colony optimization. Ad Hoc Netw 110:102317
12. Patil VS, Mane YB, Deshpande S (2019) FPGA based power saving technique for sensor node in wireless sensor network (WSN). In: Mishra BB, Dehuri S, Panigrahi BK, Nayak AK, Mishra BSP, Das H (eds) Computational Intelligence in Sensor Networks. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 385–404. https://doi.org/10.1007/978-3-662-57277-1_16
13. Chéour R, Khriji S, El Houssaini D, Baklouti M, Abid M, Kanoun O (2019) Recent trends of FPGA used for low-power wireless sensor network. IEEE Aerosp Electron Syst Mag 34(10):28–38
14. Patil V, Deshpande S (2022) Design of fpga soft core based wsn node using customization paradigm. Wireless Pers Commun 122(1):783–805
15. Toubal A, Bengherbia B, Zmirli MO, Guessoum A (2020) FPGA implementation of a wireless sensor node with built-in security coprocessors for secured key exchange and data transfer. Measurement 153:107429
16. Misra Y, Krishnaveni K, Rajasekaran AS (2022) Implementation of NLOS based FPGA for distance estimation of elderly using indoor wireless sensor networks. Mater Today: Proceed 57:2299–2306
17. Ompal MVM, Kumar A (2021) Zigbee internode communication and FPGA synthesis using mesh, star and cluster tree topological chip. Wireless Pers Commun 119:1321–1339
18. Elhosary H, Zakhari MH, Elgammal MA, Abd El Ghany MA, Salama KN, Mostafa H (2019) Low-power hardware implementation of a support vector machine training and classification for neural seizure detection. IEEE Trans Biomed Circuits Syst 13(6):1324–1337
19. Chen Y, Zhang K, Gong C, Hao C, Zhang X, Li T, Chen D (2019) T-DLA: an open-source deep learning accelerator for ternarized DNN models on embedded FPGA. In: 2019 IEEE computer society annual symposium on VLSI (ISVLSI), IEEE, pp 13–18
20. Prabakaran G, Vaithiyanathan D, Ganesan M (2022) FPGA based intelligent embedded system for predicting the productivity using fuzzy logic. Sustain Comput Inform Syst 35:100749
21. Shahrouzi SN, Perera DG (2019) Optimized hardware accelerators for data mining applications on embedded platforms: case study principal component analysis. Microprocess Microsyst 65:79–96
22. Li J, Liu M, Ma D, Huang J, Ke M, Zhang T (2020) Learning shared subspace regularization with linear discriminant analysis for multi-label action recognition. J Supercomput 76:2139–2157
23. Lambert J, Monahan R, Casey K (2021) Power consumption profiling of a lightweight development board: sensing with the INA219 and Teensy 4.0 microcontroller. Electronics 10(7):775. https://doi.org/10.3390/electronics10070775
24. Nakhkash MR, Gia TN, Azimi I, Anzanpour A, Rahmani AM, Liljeberg P (2019) Analysis of performance and energy consumption of wearable devices and mobile gateways in IoT applications. In: Proceedings of the international conference on omni-layer intelligent systems pp 68–73
25. Álvarez JL, Mozo JD, Durán E (2021) Analysis of single board architectures integrating sensors technologies. Sensors 21(18):6303

26. Girma A, Bahadori N, Sarkar M, Tadewos TG, Behnia MR, Mahmoud MN, Karimoddini A, Homaifar A (2020) IoT-enabled autonomous system collaboration for disaster-area management. IEEE/CAA J Autom Sinica 7(5):1249–1262
27. Barenghi A, Breveglieri L, Izzo N, Pelosi G (2021) Exploring cortex-M microarchitectural side channel information leakage. IEEE Access 9:156507–156527

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.