Volume (1) No (8), 2025
**Journal of Global Research in Multidisciplinary Studies (JGRMS)**
**Review Paper/Research Paper**
**Available online at** https://saanvipublications.com/journals/index.php/jgrms/index

# Low-Power Design Techniques for Embedded Processors in Portable Devices: A Review of Methods and Applications

Prof. (Dr.) Abid Hussain
*Professor*
*School of Computer Application & Technology & Dean, Research*
*Career Point University*
Kota
abid.hussain@cpur.edu.in, dean.research@cpur.edu.in

***Abstract*—Processor and system designers now priorities power consumption minimization for embedded systems that run on batteries or other portable power sources. Designing low-power embedded processors has become more important due to the increasing need for portable electronics and Internet of Things (IoT) devices. Optimization of performance and area are just as important as minimizing power consumption. As CMOS scaling advances, power consumption—both dynamic and static—poses substantial challenges in extending battery life, managing thermal constraints, and maintaining reliable operation in resource-constrained environments. Power gating, clock gating, adaptive body biasing, energy-aware software optimization, Multi-Threshold CMOS, Dynamic Voltage and Frequency Scaling (DVFS), and other low-power design strategies are reviewed in this work. It further discusses the power-performance trade-offs and emerging trends in embedded processor design, while highlighting practical applications in healthcare, automotive, consumer electronics, and industrial domains. Detailed comparisons of static versus dynamic power, processor- versus system-level consumption, and recent innovations are presented to offer holistic insight into efficient embedded system design for portable applications.**

***Keywords—Low-Power Design, Embedded Processors, Portable Devices, Energy Efficiency, Wearable Electronics.***

## I. INTRODUCTION

The growing popularity of handheld electronics and consumer electronics is a major reason why new computing parts are being designed for CMOS VLSI (very large-scale integration) systems on a chip. Optimizing battery life and extending the utility of these devices requires a revaluation of design optimizations that have historically prioritized ever-increasing performance goals and high clock rates at nearly any cost, as the emphasis moves from connected desktop computing to the mobile appliance. As the trade-offs between energy consumption and improved performance may dictate different design choices, the trend of ever-increasing complexity and size of the underlying CPU in terms of instruction issue strategies and supporting microarchitecture needs to be revaluated for these devices, as it is happening in the desktop world. Along with the conventional speed (performance) and area (cost) dimensions, power consumption emerges as a third axis in the optimization space.

Embedded processors, which serve as the core computation and control units in portable and Internet of Things (IoT) systems, must operate under strict energy constraints while supporting diverse workloads [1]. The rapid proliferation of IoT devices, excluding PCs, tablets, and smartphones has led to a staggering increase from 0.9 billion units in 2009 to over 25 billion units by 2021. Many of these devices operate in battery-powered or battery-less environments, necessitating efficient low-power design techniques.

Dynamic power and static power are two main types of embedded system power consumption. During switching operations, power is dynamically consumed, whereas static power is produced by leakage currents when the system is not in use. For battery-powered systems such as wearables and wireless sensor nodes, high power usage results in overheating and limits both usability and system lifetime. Efficient power management thus reduces thermal output, supports miniaturization, and enables longer operational periods, ultimately lowering cost, weight, and design complexity [2]. To address these challenges, various low-power design strategies have emerged. Power gating, adaptive body biasing, Dynamic Voltage and Frequency Scaling (DVFS), and other methods have recently become popular for reducing power usage. Furthermore, hardware-software co-design approaches allow software-level energy-aware decisions such as workload scheduling or core switching to directly influence hardware behaviour [3].

The need for intelligent power management has also expanded beyond component-level optimizations. In modern embedded systems, particularly those deployed in edge computing and smart environments, system-level strategies—encompassing software scheduling, memory hierarchy optimization, dataflow control, and distributed power management are now essential.

### A. *Structure of the Paper*

The structure of this paper includes: Section II, which explores embedded processors in portable devices; Section III, discussing sources of power consumption; Section IV, presenting low-power design techniques; Section V, addressing power challenges; Section VI, highlighting applications in portable embedded systems; Section VII, reviewing related literature; and Section VIII, concluding with key insights and future directions.

## II. Exploration of Embedded Processors in Portable Devices

Modern technological systems cannot function without networking and embedded processors, two interconnected but complementary subjects. Particularly designed to execute certain tasks within larger systems or devices, microcontrollers and embedded processors are specialized computing components. They are commonly used in applications that need low power consumption, small size, and real-time processing capabilities [4]. In contrast, "networking" describes the linking of various systems and devices to facilitate the transfer of information and the exchange of messages. It includes a plethora of protocols and technologies that allow for the smooth transfer of data between devices, which in turn allows for the construction of intricate networks.

A number of sectors, including consumer electronics, industrial automation, and Internet of Things (IoT) applications, have been profoundly affected by the advent of embedded processors and networking technologies. Networking systems are able to process data, maintain protocols, and perform other activities quickly by making use of the capability and computational capacity of embedded processors [5][6]. The use of embedded networking software to transmit audio and video files across wired or wireless networks is becoming increasingly common. Control applications and sensor networks are two further examples of useful uses for network capabilities. Almost every system that can connect to the Internet uses the IP protocol stack for communication.

### A. *Characteristics and Constraints of Embedded Processors in Portable Devices*

Embedded processors in portable devices are designed to perform dedicated functions within resource-constrained environments. The task-specific optimization allows for optimal performance with minimum power and area overhead, in contrast to general-purpose CPUs. Important features and limitations encompass:

#### 1) *Low Power Consumption*

Power efficiency is a primary design goal due to battery limitations in portable devices [7]. Embedded processors frequently use power-saving techniques like Dynamic Voltage and Frequency Scaling (DVFS), power gating, and sleep modes to balance performance with energy usage, ensuring longer battery life.

#### 2) *Limited Computational Resources*

Embedded processors generally have lower processing power, limited memory (RAM and ROM), and fewer peripherals compared to desktop CPUs [8]. This necessitates efficient code execution and memory management.

#### 3) *Real-Time Processing*

Many portable devices require real-time responsiveness, especially in applications like medical monitors, wearables, or control systems [9]. Embedded processors must guarantee predictable and low-latency responses to external inputs.

#### 4) *Compact Size and Integration*

Space constraints in portable devices demand compact processor designs, often as part of a SoC that integrates CPU cores, memory, I/O interfaces, and sometimes specialized accelerators (e.g., DSPs or NPUs) on a single chip.

#### 5) *Thermal Constraints*

Limited cooling options in compact devices mean that embedded processors must operate within strict thermal envelopes [10]. Excessive heat can degrade performance and reliability, making thermal-aware design critical.

#### 6) *Application-Specific Optimization*

Embedded processors are often tailored to the specific needs of the application, such as low-latency audio processing in earbuds or sensor data fusion in fitness trackers. This specialization enhances efficiency but limits flexibility.

#### 7) *Reliability and Stability*

Since many portable systems operate in critical or autonomous roles (e.g., health devices or industrial sensors), embedded processors must maintain high reliability under varying environmental conditions and extended operation.

### B. *Challenges in Low-Power Processor Design*

Powering a SoC from a battery presents several challenges, Battery-operated SoCs must ensure minimal energy usage to prolong battery life, requiring efficient hardware and workload optimization as energy efficiency must be carefully balanced with key system attributes such as performance, reliability, and scalability, as illustrated in Figure 1. These challenges include:

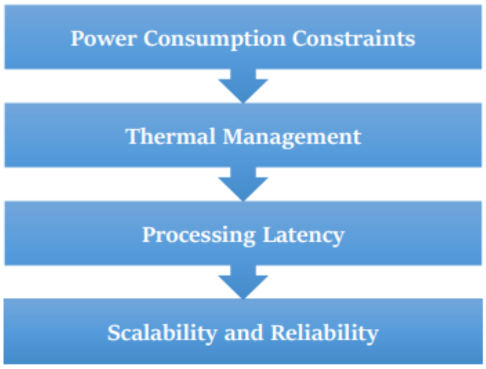


Fig. 1. Challenges in Low-Power Processor Design

- **Power Consumption Constraints:** IoT devices, along with embedded systems, are constrained in power supply [11], including battery or energy harvesting. A significant issue that must be addressed is efficient computation with low power consumption because excessive power consumption reduces battery lifespan and leads to more frequent charges.
- **Thermal Management:** Low-power processors realize lower power consumption, but converting power into heat poses a reliability and systems stability problem. Thermal regulation is crucial to prevent the components from overheating and not use cooling products like fans.
- **Processing Latency:** A common set of applications include real-time artificial intelligence computations, robotics, smart sensors, etc., which demand low latency data processing at the same time as low energy consumption. Energy savings should.
- **Scalability and Reliability:** Low-power processors have been required for their scalability and flexibility in various types of work while having sufficient dependability. To elaborate [12], with International

Journal for Multidisciplinary Research ever-growing levels of interconnectivity in devices.

### C. Application Domains of Embedded Processors in Portable Devices

Embedded processors play a pivotal role in a wide range of portable devices, offering targeted performance with minimal power consumption. Their versatility and efficiency make them ideal for various application domains where compact size, energy efficiency, and task-specific processing are essential. Key application domains include:

#### 1) Smartphones and Tablets

Embedded processors manage a variety of tasks, from handling user interfaces and sensors to managing wireless communications and multimedia [13]. Modern mobile SoCs integrate CPU, GPU, AI accelerators, and modem functions into a single chip for optimal performance and power efficiency.

#### 2) Wearable Devices

Wearables such as smartwatches, fitness bands, and health trackers rely on ultra-low-power embedded processors to continuously monitor biometric data (e.g., heart rate, activity, sleep) and communicate wirelessly with other devices, often operating under stringent battery constraints.

#### 3) Internet of Things (IoT) Devices

IoT applications ranging from home automation systems to industrial sensors leverage embedded processors for localized data processing, energy-efficient communication, and autonomous decision-making [14]. Processors in these devices are optimized for low-power wireless connectivity and event-driven operation.

#### 4) Medical and Healthcare Devices

Portable medical instruments such as insulin pumps, ECG monitors, portable ultrasound devices, and smart inhalers use embedded processors for real-time data acquisition, signal processing, and secure data transmission [15]. Reliability, safety, and low power are critical in these domains.

#### 5) Consumer Electronics

Devices like portable gaming consoles, digital cameras, e-book readers, and Bluetooth speakers incorporate embedded processors to manage multimedia processing, user interactions, and connectivity while maintaining lightweight and compact form factors.

#### 6) Automotive Systems

Modern vehicles include portable diagnostic tools and infotainment systems powered by embedded processors. These devices must offer reliable, real-time performance and robustness in dynamic environments.

#### 7) Industrial and Field Equipment

Embedded processors are used in handheld diagnostic tools, rugged portable data loggers, and remote sensors for industrial monitoring and maintenance. These systems require high reliability, real-time processing, and energy efficiency in harsh conditions.

#### 8) Educational and Assistive Technologies

Devices like electronic dictionaries, language translators, portable projectors, and assistive communication tools (e.g., AAC devices) utilize embedded processors for interactive, responsive performance in lightweight, mobile formats.

## III. Sources of Power Consumption in Embedded Systems

### A. Static and Dynamic Power Consumption

Power consumption in CMOS circuits is mostly made up of static power and dynamic power. Leakage currents, which are inherent to any active circuit and contribute to static power consumption or leakage power, exist regardless of clock rates or usage circumstances. This static power is mostly dictated by the process technology and kind of transistors [16]. Because it necessitates changes to the system's architecture at the lowest level, lowering the static power is outside the scope of this chapter. Further information about potential strategies to enhance energy efficiency.

The key factors that determine dynamic power consumption, which is generated by circuit activity (such as transistor switches and register value changes), are the specific use case, clock rates, and input/output (I/O) operations [17][18]. The dynamic power consumption is caused by switching capacitance and short-circuited current. Although short-circuit current accounts for a small percentage of overall power consumption (around 10-15%), no viable solutions have been identified to lower this figure without sacrificing performance. Since switched capacitance is the main contributor to dynamic power consumption, and can be described in Equation (1):

$$P_{dynamic} = aCV^2f \qquad (1)$$

The physical capacitance (C), the supply voltage (V), the clock frequency (f), and the switching activity (a). A system's low-level architecture dictates its switching activity and capacitance levels. Table I provides a comparative analysis of static and dynamic power consumption discussed below:

TABLE I. Comparison of Static vs. Dynamic Power Consumption

| Aspect | Static Power Consumption | Dynamic Power Consumption |
|---|---|---|
| Cause | Leakage currents in transistors | Transistor switching and circuit activity |
| Dependency | Independent of clock rate and usage | Depends on clock rate, switching activity, and I/O usage |
| Main Contributor | Subthreshold and gate leakage | Switched capacitance (dominant), short-circuit current |
| Reduction Techniques | Power gating, low-leakage transistors | Clock gating, DVFS, reducing switching activity |
| Design Level Control | Controlled at transistor/fabrication level | Controlled at architectural and software levels |

### B. Processor-Level vs. System-Level Consumption

In portable embedded systems, efficient power management is critical due to limited energy resources and stringent thermal constraints. Power consumption can be broadly categorized into two main domains: processor-level and system-level. Understanding these domains separately is essential for implementing targeted low-power design strategies.

Processor-level power consumption refers to the energy used by the internal components of the embedded processor, including the arithmetic logic unit (ALU), control logic, registers, internal buses, and on-chip cache memory. The two primary contributors to this domain are dynamic power and static (leakage) power [19]. Dynamic power is consumed during transistor switching activities and is a function of switching frequency, load capacitance, and supply voltage. Table II presents a comparison between processor-level and system-level power consumption, highlighting their Definition, Main Components, Power Types, optimization scope and control techniques:

TABLE II. COMPARISON OF PROCESSOR-LEVEL VS. SYSTEM-LEVEL POWER CONSUMPTION

| Aspect | Processor-Level Consumption | System-Level Consumption |
|---|---|---|
| Definition | Power consumed by internal processor components (ALU, registers, cache) | Power consumed by all system components including processor, memory, I/O, etc. |
| Main Components | ALU, control logic, internal buses, on-chip cache | Display, sensors, external memory, I/O interfaces, communication modules |
| Power Types | Dynamic (switching) and static (leakage) power | Includes processor power plus peripheral and interconnect-related power |
| Optimization Scope | Circuit-level and microarchitectural optimizations | System-level design, software scheduling, hardware-software co-design |
| Control Techniques | DVFS, clock gating, transistor sizing | Power-aware scheduling, component duty cycling, interface management |

## IV. LOW-POWER DESIGN TECHNIQUES

### A. Voltage Scaling (DVFS)

One of the most effective strategies for lowering power consumption in digital circuits is voltage scaling, specifically Dynamic Voltage and Frequency Scaling (DVFS). Reducing the voltage can have a significant impact on power dissipation because dynamic power is directly proportional to the square of the supply voltage. In order for DVFS to function, the system's workload determines the operating frequency and supply voltage, which are dynamically adjusted [20]. When performance demands are low, the processor operates at reduced voltage and frequency, conserving energy. Conversely, under high workloads, voltage and frequency are increased to maintain system performance. In embedded and mobile systems, this method is commonly employed due to the crucial need of energy efficiency.

### B. Power Gating

One method to decrease power leakage is power gating, which involves turning off power to inactive functional units or blocks on a chip. To turn off the power while the block isn't in use, it employs sleep transistors typically high-threshold voltage transistors, placed between the supply voltage and the logic circuit. This significantly reduces static power caused by leakage currents. Power gating is effective during long idle periods and is especially useful in SoC (System-on-Chip) designs where different modules operate independently. However, it requires careful management of wake-up times and state retention to ensure seamless transitions between active and sleep modes.

### C. Clock Gating

A dynamic power reduction technique known as clock gating is used to deactivate the clock signal to non-active parts of a circuit. Since the clock network contributes significantly to switching power due to its constant activity, gating the clock saves considerable power by preventing unnecessary toggling. It is implemented by adding enable signals to the clock path, ensuring that only the required parts of the circuit receive the clock signal during operation. Clock gating is commonly integrated into RTL designs and is supported by many synthesis tools. It offers a balance between design complexity and power savings, making it a staple in modern low-power VLSI design.

### D. Multi-Threshold CMOS (MTCMOS)

Multi-Threshold CMOS (MTCMOS) technology uses transistors with various threshold values in a single design to create the best power and performance. Transistors with a low threshold voltage are utilized in timing-critical circuits to keep speed constant, whilst those with a high threshold voltage are utilized in non-critical paths to decrease leakage current [21]. This dual-threshold approach enables designers to reduce leakage power without compromising overall system performance. MTCMOS is particularly effective in sub-micron technologies where leakage becomes more prominent. The challenge lies in determining the right threshold voltages and transistor placements, which often require advanced synthesis and place-and-route tools.

### E. Adaptive Body Biasing (ABB)

The voltage given to the body (or substrate) of transistors can be changed during runtime using adaptive body biassing. Forward body biasing lowers the threshold voltage to increase performance, while reverse body biasing raises it to reduce leakage during standby periods [22]. ABB provides a fine-grained control over leakage and performance trade-offs, especially useful in sub-threshold and near-threshold designs. It allows systems to adapt to variations in process, voltage, and temperature (PVT) conditions dynamically. ABB is often implemented using on-chip bias generators and is complementary to other low-power techniques like DVFS and MTCMOS.

### F. Energy-Aware Compilation and Software Optimization

Energy-aware compilation and software optimization involve modifying the code and compiler behavior to generate energy-efficient binaries. Techniques include minimizing memory access, reducing control overhead, efficient instruction scheduling, loop unrolling, and power-aware register allocation. At a higher level, application profiling and dynamic code adaptation can be employed to tailor software behavior to the energy constraints of the hardware. Compilers can also exploit hardware features such as DVFS interfaces or sleep modes in processors [23]. By considering energy consumption during software development, significant power savings can be achieved, especially in embedded systems where both hardware and software are tightly coupled.

## V. POWER CHALLENGES IN EMBEDDED PROCESSORS FOR PORTABLE DEVICES

Embedded processors in portable devices are critically challenged to power due to the conflicting and inconsistent performance, size and battery life expectations. As the technology shrinks further with improved functionalities, power density increases. Power dissipation typically increases with power density, leading to increased thermal output and energy consumption. Dynamic power, which can be described as the switched activity, and static power, due to leakage currents, add to the total power dissipation, both of which are increased when the technology is in the deep sub-micron range. Portable devices and gadgets are also subjected to a very strict energy budget that limits functionality under reasonable power consumption. Therefore, low-power design approaches such as Dynamic Voltage and Frequency Scaling (DVFS), clock gating, power gating and architectural enhancements, Low-power techniques such as DVFS, power gating, and energy-aware software optimization help manage energy efficiently while maintaining performance and reliability. These methods are crucial for IoT systems, which exclude PCs, tablets, and smartphones, and increasingly operate in battery-powered or battery-less environments, as shown in Figure 2.

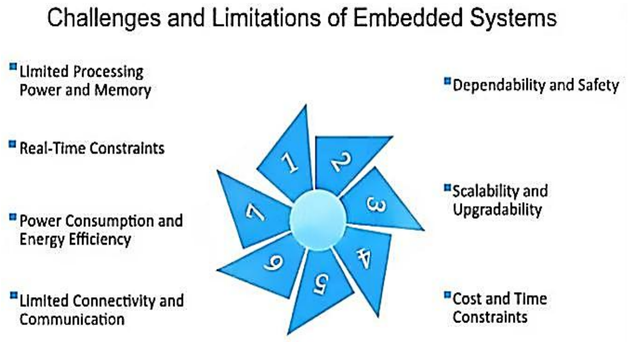


Fig. 2. Challenges in Embedded Systems

### *A. Power Constraints in Mobile and Wearable Environments*

Pervasive biomedical applications like gesture-based control, health monitoring, and activity tracking can be made possible with well-designed, small-form-factor wearable devices that won't sacrifice user experience. But usually, smaller form factors have lower battery capacities, which means have to charge and replace the battery more often than like. While smaller and lighter flexible batteries have obvious benefits, their capacities still fall short of what wearable devices need for smooth operation [22]. Success for wearable IoT devices depends on making the most of limited energy resources while maximizing utilization, or productive work. The power consumption of the sensors, microcontroller, and BLE was first measured and characterized independently during gesture recognition. This paper's thorough energy characterization allowed us to create a new compact energy model that optimization techniques can employ during runtime.

### *B. Trends in Energy Efficiency*

The environmental, social, and financial spheres can all reap the rewards of energy efficiency. Despite progress in energy efficiency, the benefits are being largely ignored as a result of the general uptick in economic activity. Greenhouse gas emissions may be reduced, energy sources could be diversified, and reliance on fossil fuel markets could be diminished through the use of renewable energy [23]. Green technology innovation, which renewable energy sources can spur, might potentially increase job opportunities in the European Union.

### *C. Power-Performance Trade-Offs*

The integration of designs in terms of power and performance is becoming more challenging. If conducting extensive design space exploration early on, may place design in the best possible subspace for better convergence and to save money on redesigns down the road. Leveraging or re-optimizing existing designs is becoming more popular in today's quickly changing industrial landscape. Massive reductions in costs are achievable with leveraged or re-optimized designs due to their laser-like focus on improving performance, decreasing power consumption, or achieving both goals. Sometimes extra design choices are required, but sometimes scaling technology can accomplish some of these aims. Compared to ground-up designs, parameters in system-level models have a greater confidence level, making sophisticated design exploration methodologies and tools ideal for leveraged designs. By coming up with several design choices for each block and then exploring the solution space with tools like the one presented in this article, design engineers can go ahead and avoid redesigns later on in the design process, which ensure a faster time to market.

## VI. Applications in Portable Embedded Systems

Personal health, safety, and lifestyle is being transformed by portable embedded systems using compact low-power devices that include wearable, fitness tracking and IoT sensor nodes. These systems would be continuously able to collect, process and transmit physiological or environment data with high energy efficiency and are therefore adroit to be used in real-life environment over long-term observations.

### *A. Wearables and Fitness Trackers*

The term "wearables" describes a category of smart consumer products that are quickly becoming standard issue. These devices track digital health data. People are becoming more health conscious, and this reflects that. For both quick and thorough diagnosis of a variety of diseases, wearable biosensors may continually measure a person's physiological data in real time; these devices are non-invasive, non-irritating, and inexpensive. Individualized patient health monitoring is another benefit they provide. Devices that can be worn by the user come in a variety of shapes and sizes, and they can be fastened to various portions of the body or worn on clothing. They can be classified as head, limb (including arms), leg, eye, or torso wearable devices according to their point of contact. They can also be classified as skin-based or biofluid-based depending on the kind of probe they use. Wearable intelligent insoles for diabetic foot monitoring, gadgets for real-time detection of heart attacks, and smart-digital stethoscope systems are only a few examples of the specialized monitoring wearable devices that are available. Clinicians frequently recommend its usage. Consumer gadgets that are skin-based and worn on the wrist and offer continuous data for the detection of various diseases have been the primary focus of this review [24].

### *B. Iot Sensor Nodes and Biomedical Devices*

An integral part of a wireless body area network for monitoring the human body are wearable sensor nodes. They can be utilized for collecting environmental variables surrounding the human body, for example in safety applications, in addition to medical signals. A Bluetooth wireless interface is provided by Reference as part of a wearable sensor system for monitoring volatile organic chemicals that are harmful. For the Internet of Things, there are a few cellular-free options [25]. With little power consumption, offer the greatest receiver sensitivity and the largest cover distance.

## VII. Literature of Review

This section highlights key innovations in low-power embedded systems, including low-energy processor design, multi-core architectures, and real-time performance optimization. It also emphasizes practical applications in healthcare, automotive, and industrial domains, aiming to deliver compact, reliable, and energy-efficient solutions for modern portable devices discuss in Table III.

 

Okugawa and Mouri (2025) presented Large-scale embedded systems are increasing in performance and multifunctionality, so the increase in the number of hardware causes problems such as weight increase and internal space compression. In addition, while multi-core processors are becoming popular in embedded devices, existing embedded software does not take full advantage of them. To solve these problems, propose a method to aggregate multiple existing embedded software into a single hardware with multi-core processors. Virtualization technology is generally used when aggregating multiple software, but it is difficult to guarantee real-time performance and porting cost for aggregation with existing technology [26].

Yang and Gang (2025) presented current situation that various kinds of real-time communication protocols and real-time control systems running on modern industrial control equipment need many types of high-precision timing services, a design method of timer based on Free RTOS real-time embedded operating system is proposed. Using the deterministic real-time task management mechanism of Free RTOS real-time embedded operating system, the timing service addition, timer counting, and timing service function operation are managed by special thread tasks, which solves the problem of large cumulative error caused by the deep coupling of the above operations [27].

Aalund and Philip Paglioni (2025) carried out Hardware dependability is very important for embedded systems because they are used in harsh environments and last a long time. Further, embedded systems' uses can be mission-or safety-critical, with severe financial consequences in the case of a malfunction. Such applications and environments necessitate a high level of dependability. The difficulties of attaining hardware reliability in embedded systems are examined in this literature review. It summarizes hardware reliability completely by analyzing important studies from various perspectives and techniques. Embedded system hardware failure mechanisms are examined in this study, along with mitigation measures and new trends that shape embedded system design in the future [28].

Zhang, Zhang and Feng (2024) explored low-power microprocessor design and optimization techniques, focusing on how to effectively reduce power consumption without significantly affecting performance. with an overview of the fundamentals and major challenges of low-power design, followed by an in-depth analysis of a variety of optimization techniques at the architectural level, circuit level, and algorithmic and software levels. This paper demonstrates the effects and trade-offs of these techniques in practical applications through specific case studies, and provides an outlook on future trends and research directions in low-power microprocessor design [29].

Alves et al. (2024) improved safety and efficiency of land vehicles, such as cars and trucks, frequent tire inspections are necessary. Regular checks of tire tread depths help prevent accidents and enhance fuel efficiency. However, traditional methods for estimating tread depth, like depth gauges, can be both time-consuming and imprecise, especially for vehicles with multiple tires. In this paper, propose a novel portable embedded Digital Processing System (DPS) device that utilizes linear image sensors (CCDs) and active illumination to estimate the depths of all tire treads at once. The device features a microcontroller that configures the linear sensor's parameters and collects its readings [30].

Marimuthu, Shanuja and Aparna (2024) conducted the need of continuous health monitoring for ICU patients, persons with chronic disease, elder people, home alone people etc. Wireless technology, Body Area Networks using smart devices, artificial intelligence provides the freedom for the patients/persons to be continuously monitored at any place and at any time. Portable IoT smart wearable gadgets are firmly recommended in dynamic and complex real-world health issues. In medical field, all the wearable gadgets are highly used due to its intelligence and data handling capability. Today's bio smart devices help to reduce the risk in diagnosis or treatment; it provides early diagnosis, long term continuous health monitoring, rehabilitate illnesses, and the person can be done medication irrespective of the location where he/she resides [31].

Table III provides a comparative overview of key studies on low-power embedded systems, summarizing their research focus, methodological approaches, principal findings, identified challenges, and suggested directions for future work.

TABLE III. COMPARATIVE ANALYSIS OF LITERATURE BASED ON LOW-POWER EMBEDDED SYSTEMS AND APPLICATIONS.

| Reference | Study Focus | Approach | Key Findings | Challenges | Future Directions |
|---|---|---|---|---|---|
| Okugawa and Mouri (2025) | Embedded software integration on multi-core processors | Aggregation of multiple embedded software using multi-core processors; avoids conventional virtualization | Reduces hardware size and enhances performance of embedded systems | Difficulty in ensuring real-time performance and high porting cost with existing virtualization | Develop efficient aggregation frameworks ensuring real-time guarantees with lower porting costs |
| Yang and Gang (2025) | Real-time timer design in industrial control systems | Timer design using FreeRTOS real-time OS and task-based thread management | Solves timing service errors and reduces cumulative timing drift | Deep coupling of timing operations leads to error accumulation | Enhanced timer accuracy and integration in complex real-time industrial systems |
| Aalund and Paglioni (2025) | Hardware reliability in embedded systems | Literature survey analyzing failure modes and reliability strategies | Highlights importance of reliability in extreme and long-term embedded applications | Ensuring high reliability in harsh, long-term environments | Adopt predictive maintenance and robust hardware design methodologies |
| Zhang, Zhang, and Feng (2024) | Low-power microprocessor design and optimization | Multilevel optimization (architectural, circuit, algorithmic/software levels) with case studies | Demonstrates effective power reduction with minimal performance tradeoffs | Balancing power efficiency with required performance and complexity | Advance cross-layer optimization methods and explore AI-based low-power design |
| Alves et al. (2024) | Tire tread depth estimation for vehicle safety | Portable embedded DPS using CCD sensors and active illumination | Enables accurate and fast measurement for all tires simultaneously | Traditional methods are slow and inaccurate; sensor integration complexity | Expand portable DPS for broader vehicle diagnostics and commercial deployment |

| | | | | | |
|---|---|---|---|---|---|
| Marimuthu, Shanuja, and Aparna (2024) | Health monitoring using wearable IoT in healthcare | Smart wearable sensors with AI and wireless communication (Body Area Networks) | Enables real-time, remote, and continuous health monitoring | Need for reliable connectivity, data handling, and privacy | Advance AI-driven diagnostics and integrate cloud-based health monitoring systems |

## VIII. CONCLUSION AND FUTURE WORK

Low-power embedded processor design has become an essential area of focus due to the proliferation of battery-powered and portable computing systems, especially within IoT and wearable device domains. This paper reviewed key power-saving methodologies and outlined their implementation challenges, architectural considerations, and real-world relevance. Techniques like DVFS, power gating, and MTCMOS have shown significant potential in balancing energy efficiency with system performance. Additionally, energy-aware compilation and adaptive body biasing offer promising pathways for software-hardware co-optimization. Future work involve deeper integration of AI-driven energy management, context-aware power control, and advanced system-level modelling to enhance energy predictability. Moreover, as application domains grow increasingly heterogeneous, cross-layer design strategies combining device-level, architectural, and software-level optimizations will be pivotal in pushing the boundaries of low-power embedded systems.

## REFERENCES


[1] W. Ma, Q. Cheng, Y. Gao, L. Xu, and N. Yu, “An Ultra-Low-Power Embedded Processor with Variable Micro-Architecture,” *Micromachines*, vol. 12, no. 3, p. 292, Mar. 2021, doi: 10.3390/mi12030292.

[2] S. Mittal, “A survey of techniques for improving energy efficiency in embedded computing systems,” *Int. J. Comput. Aided Eng. Technol.*, vol. 6, no. 4, p. 440, 2014, doi: 10.1504/IJCAET.2014.065419.

[3] H. Cherupalli, H. Duwe, W. Ye, R. Kumar, and J. Sartori, “Determining Application-specific Peak Power and Energy Requirements for Ultra-low Power Processors,” *ACM Trans. Comput. Syst.*, vol. 45, no. 1, pp. 3–16, Apr. 2017, doi: 10.1145/3037697.3037711.

[4] R. Patel, “Security Challenges in Industrial Communication Networks: A Survey on Ethernet/Ip, Controlnet, and Devicenet,” *Int. J. Recent Technol. Sci. Manag.*, vol. 7, no. 8, pp. 54–63, 2022, doi: 10.10206/IJRTSM.2025171772.

[5] M. D. Jaraíz-Simón, J. A. Gómez-Pulido, M. A. Vega-Rodríguez, and J. M. Sánchez-Pérez, “Fast Decision Algorithms in Low-Power Embedded Processors for Quality-of-Service Based Connectivity of Mobile Sensors in Heterogeneous Wireless Sensor Networks,” *Sensors*, vol. 12, no. 2, pp. 1612–1624, Feb. 2012, doi: 10.3390/s120201612.

[6] A. Goyal, “Integrating Blockchain for Vendor Coordination and Agile Scrum in Efficient Project Execution,” *Int. J. Innov. Sci. Res. Technol.*, vol. 9, no. 12, 2024.

[7] K. J. Kim *et al.*, “Six Strategies for E ff ective Learning,” *Handb. Self-Regulation Learn. Perform.*, 2015.

[8] H. S. Chandu, “Enhancing Manufacturing Efficiency: Predictive Maintenance Models Utilizing IoT Sensor Data,” *IJSART*, vol. 10, no. 9, 2024.

[9] Y. K. Raut *et al.*, “An embedded system — a review paper,” *Int. J. Adv. Sci. Res. Eng. TRENDS*, vol. 8, no. 12, pp. 9–13, 2024.

[10] M. Yadav, “The Role of Embedded Systems in Robotics A Comprehensive Review,” *Int. J. Eng. Res. Technol.*, vol. 14, no. 03, 2025.

[11] V. Panchal, “Thermal and Power Management Challenges in High-Performance Mobile Processors,” vol. 13, no. 11, 2024, doi: 10.15680/IJIRSET.2024.1311014.

[12] S. Mittal, “A survey of architectural techniques for improving cache power efficiency,” *Sustain. Comput. Informatics Syst.*, vol. 4, no. 1, pp. 33–43, Mar. 2014, doi: 10.1016/j.suscom.2013.11.001.

[13] Y. Lu, “The application of embedded system on electric devices,” *J. Phys. Conf. Ser.*, vol. 2649, no. 1, 2023, doi: 10.1088/1742-6596/2649/1/012030.

[14] T. S. Ajani, A. L. Imoize, and A. A. Atayero, “An Overview of Machine Learning within Embedded and Mobile Devices–Optimizations and Applications,” *Sensors*, vol. 21, no. 13, Jun. 2021, doi: 10.3390/s21134412.

[15] V. Rajavel, “Integrating Power-Saving Techniques into Design for Testability of Semiconductors for Power-Efficient Testing,” vol. 07, no. 243, pp. 243–251, 2025, doi: 10.37547/tajet/Volume07Issue03-22.

[16] J. Zidar, T. Matić, I. Aleksi, and Ž. Hocenski, “Dynamic Voltage and Frequency Scaling as a Method for Reducing Energy Consumption in Ultra-Low-Power Embedded Systems,” *Electron.*, 2024, doi: 10.3390/electronics13050826.

[17] I. Ali, Y.-I. Jo, S. Lee, W. Y. Lee, and K. H. Kim, “Reducing Dynamic Power Consumption in Mixed-Critical Real-Time Systems,” *Appl. Sci.*, vol. 10, no. 20, Oct. 2020, doi: 10.3390/app10207256.

[18] V. Panchal, “Energy-Efficient Core Design for Mobile Processors : Balancing Power and Performance,” *Int. Res. J. Eng. Technol.*, vol. 11, no. 12, pp. 191–201, 2024.

[19] V. Shah, “Next-Gen Emergency Communication Using LowPower Wide-Area and Software-Defined WANS,” *Int. J. Adv. Res. Sci. Commun. Technol.*, vol. 2, no. 1, pp. 600–609, 2022, doi: 10.48175/IJARSCT-8349M.

[20] D. K. Sharma and R. Vikram, “Multi-Voltage Design of RISC Processor for Low Power Application: A Survey,” *Int. J. Comput. Digit. Syst.*, vol. 15, no. 1, pp. 1019–1035, 2024, doi: 10.12785/ijcds/150172.

[21] J. Li, Y. Lin, N. Ning, and Q. Yu, “A +0.44°C/−0.4°C Inaccuracy Temperature Sensor With Multi-Threshold MOSFET-Based Sensing Element and CMOS Thyristor-Based VCO,” *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 68, no. 3, pp. 1102–1113, Mar. 2021, doi: 10.1109/TCSI.2020.3042825.

[22] J. Park, G. Bhat, A. NK, C. S. Geyik, U. Y. Ogras, and H. G. Lee, “Energy per Operation Optimization for Energy-Harvesting Wearable IoT Devices,” *Sensors*, vol. 20, no. 3, Jan. 2020, doi: 10.3390/s20030764.

[23] V. Kolev, I. Draganova-Zlateva, and D. Gospodinova, “Trends in Energy efficiency and CO 2 emissions according to Bulgarian National Energy Efficiency plan,” in *2019 11th Electrical Engineering Faculty Conference (BulEF)*, IEEE, Sep. 2019, pp. 1–6. doi: 10.1109/BulEF48056.2019.9030770.

[24] S. Chakrabarti, N. Biswas, L. D. Jones, S. Kesari, and S. Ashili, “Smart Consumer Wearables as Digital Diagnostic Tools: A Review,” *Diagnostics*, vol. 12, no. 9, Aug. 2022, doi: 10.3390/diagnostics12092110.

[25] F. Wu, C. Rudiger, J.-M. Redoute, and M. R. Yuce, “WE-Safe: A wearable IoT sensor node for safety applications via LoRa,” in *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, IEEE, Feb. 2018, pp. 144–148. doi: 10.1109/WF-IoT.2018.8355234.

[26] K. Okugawa and K. Mouri, “Homare: Platform Software for Aggregating Embedded Systems on Multi-core Processor,” in *2025 8th International Conference on Information and Computer Technologies (ICICT)*, IEEE, Mar. 2025, pp. 35–42. doi: 10.1109/ICICT64582.2025.00012.

[27] J. Yang and M. Gang, “Design Method of Timer Based on FreeRTOS Real-Time Embedded Operating System,” in *2025 4th International Symposium on Computer Applications and Information Technology (ISCAIT)*, IEEE, Mar. 2025, pp. 1911–1916. doi: 10.1109/ISCAIT64916.2025.11010702.

[28] R. Aalund and V. Philip Paglioni, “Enhancing Reliability in Embedded Systems Hardware: A Literature Survey,” *IEEE Access*, vol. 13, pp. 17285–17302, 2025, doi: 10.1109/ACCESS.2025.3534138.


 

[29] Y. Zhang, S. Zhang, and L. Feng, "Research on low-power microprocessor design and optimization technology," in *2024 International Conference on Power, Electrical Engineering, Electronics and Control (PEEEC)*, 2024, pp. 362–366. doi: 10.1109/PEEEC63877.2024.00072.

[30] D. N. A. Alves, A. dos Santos, B. G. Ferreira, and T. F. Vieira, "Portable Device for Faster Tire Tread Depth Estimation Using Linear CCD Sensors," in *2024 XIV Brazilian Symposium on Computing Systems Engineering (SBESC)*, IEEE, Nov. 2024, pp. 1–6. doi: 10.1109/SBESC65055.2024.10771924.

[31] P. Marimuthu, C. C. Shanuja, and N. Aparna, "Portable IoT Smart Devices in Healthcare and Remote Health Monitoring – Abnormality Detection through Personalized Vital Health Signs Using Smart Bio Devices," in *Internet of Things in Bioelectronics*, Wiley, 2024, pp. 201–218. doi: 10.1002/9781394241903.ch10.