A review on Power optimization & Energy Efficient Techniques for Embedded systems

Lahari R, Lohith T N, Manjunath Mandi, Govinda Raju M
Department of ECE, RV College of Engineering, Bengaluru

***Abstract*— As embedded systems continue to play a critical role in various industries, optimizing power consumption has become a key challenge. This paper explores recent advancements in low-power design methodologies, energy harvesting technologies, and power optimization techniques for embedded systems. We present a comprehensive overview of hardware and software strategies for low-power design methodology, innovative energy harvesting approaches such as solar energy, thermal energy ,RF energy and vibration and mechanical harvesting , and power improved techniques such as Dynamic Voltage and Frequency Scaling (DVFS), power gating, clock gating and PSoC. These innovations enable sustainable, energy- efficient embedded systems suitable for the next generation of IoT and mobile devices.**

I. INTRODUCTION

Embedded systems play a pivotal role in modern computing, finding applications in consumer electronics, automotive systems, industrial automation, and healthcare devices. As the reliance on these systems increases, energy efficiency has become a primary design consideration due to the growing demand for battery-operated devices, environmental concerns, and thermal management challenges. The increasing complexity and functionality of embedded systems have further intensified the need for energy-efficient solutions. To address these challenges, researchers have explored various techniques aimed at optimizing power consumption at different levels of embedded system design. This paper provides a comprehensive analysis of latest developments in low- power design methodologies, energy harvesting, and power optimization techniques, focusing on strategies across hardware, software, and architectural domains. We analyze recent advancements in the field to offer insights into current trends and future directions in low-power design methodologies, energy harvesting, and power optimization techniques in embedded system design. Additionally, we are reviewing voltage scaling strategies for ultra-low-power embedded systems. These methods optimize energy efficiency by dynamically adjusting voltage levels based on workload demands, reducing unnecessary power consumption.

This review aims to provide a holistic understanding of energy-efficient embedded system design, power optimization techniques and low power design methodologies, synthesizing findings from recent literature to guide researchers and practitioners in developing future solutions

II. LITERATURE SURVEY

[1] Designing power-efficient embedded systems with Programmable System on Chip(PSoC) is reviewed in this work, which serves as a initial point to understand power known design strategies. A very high degree of flexibility and configurability is offered by PSoC, a truly programmable embedded SoC (System on Chip) that combines memory, a microprocessor, and programmable analog and digital components onto a single chip.

[2] The new low power features of modern microcontrollers—in particular, deep sleep technology—are reviewed in this article. Additionally, it goes over the components of power consumption and how to set up the microcontroller to obtain the highly low power essential for an embedded system.

[3] Given the features and architectures of the memory devices, this study suggests a segment-aware energy- efficient management to increase power efficiency. In the suggested method, if the power consumption is thought to be lower, the program code is moved from one memory device to another. By using the approach, we also examine and asses the overall impacts on energy efficiency. Our model uses less electricity than the computer code that hasn't been altered.

[4] The effectiveness of the MTCMOS technique in designing power-efficient counters for modern low-power applications, including wearable electronics, embedded systems, and Internet of Things devices, is demonstrated in this paper. The proposed solution provides a practical and scalable framework for handling power problems in digital sequential circuits.

[5] This research examines and evaluates a number of optimization strategies for lowering power usage. A power management controller is created as part of the experiment to regulate and track the peripherals' power usage. DFS, Clock gating and power down mode are shown with the results.

[6] In this study, we examine methods for controlling embedded systems' power usage. We go over the necessity of power management and categorize the methods based on a number of crucial factors to show their parallels and divergences. The goal of this study is to assist researchers and application developers in better understanding how power management strategies operate and creating even more effective high-performance embedded systems.

III. LOW POWER DESIGN METHODOLOGIES

1. Hardware Optimization

A. Low-Power Microcontrollers (MCUs)

Many modern MCUs offer multiple power modes (e.g., active, sleep, deep-sleep) to adapt to different operational needs. ARM Cortex-M processors support power-saving states like deep sleep or dynamic frequency scaling to reduce energy consumption during idle periods. Historically, apps have achieved lower power consumption by putting the MCU into sleep mode. To counteract the negative consequences of growing complexity and smaller process geometries, the manufacturers have incorporated new modes into their microcontrollers. We'll call these modes "deep sleep" [2]. They may also be known as LPM5, standby, STOP2, or deep sleep. Although the amount of improvement brought about by deep sleep varies from manufacturer to manufacturer, an 80% decrease in sleep current is typical. Indeed, in deep-sleep mode, some MCUs can currently drain as little as 20 nA. Deep sleep can extend an application's lifespan by years by combining low currents with batteries that have low rates of self-discharge [7].

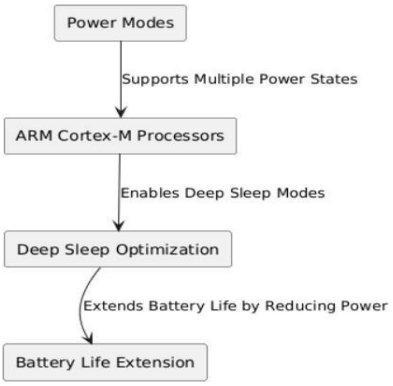

Fig 1. Low Power MCUs

B. Low-Leakage Components:

It refers to electronic components designed to minimize leakage current, which is a small current that continues to flow even when a circuit is not actively switching, thereby reducing power consumption when the device is in idle or sleep mode, crucial for battery-powered applications. Leakage current, which flows even when components are in a standby state, can significantly contribute to power consumption. Utilizing low-leakage components is a key strategy to reduce this issue. Selecting components with lower leakage current can drastically reduce standby power consumption. Voltage regulators and low-power memories are also critical in reducing total system power. Low-leakage Transistors: Advanced semiconductor technologies offer transistors with reduced leakage characteristics. These transistors are particularly beneficial in devices that spend extended periods in standby mode. Low-Power Memories: Memory components, such as SRAM and DRAM, can be significant contributors to energy consumption. Selecting low-power memory options with reduced leakage and efficient refresh cycles is essential for minimizing overall system power.

Offloading tasks to dedicated co-processors (e.g., Digital Signal Processors or Hardware Accelerators) reduces the load on the central processor, minimizing overall energy consumption. Peripherals and subsystems can account for a significant portion of a system's energy consumption. Optimizing these components can lead to substantial energy savings.

C. Energy efficient peripheral design

In the context of embedded system hardware

optimization refers to designing peripheral components (like ADCs, DACs, UARTs, etc.) within an embedded system with a focus on minimizing power consumption, of may achieved from techniques like clock gating, power gating, optimized circuit topologies, and selecting low-power memory technologies, allowing the system to operate efficiently on battery power or with reduced energy usage.

2. Software Optimization

While hardware plays a critical role, software optimization is equally essential in low-power design methodologies. Software techniques like efficient code execution, power- aware scheduling, and adaptive algorithms can complement hardware strategies to achieve optimal energy efficiency.

A.Power Management System for Embedded RTOS

Applications or operating systems (OS) will be used to create power management systems for embedded devices. Developers can focus solely on creating applications if the OS's power management strategy is implemented. OS has precise and detailed information on the many tasks that have been carried out. Consequently, it makes sense to adopt algorithms that assign lower power states to components that are not in use. This can drastically lower the system's energy consumption. The real-time operating system (RTOS) provides a comprehensive set of power management application programming interfaces (APIs) for both device drivers and programs within a power management component. This paper presents abstracted concepts of device power managers, application power managers, and system power managers (PM) [13]. It illustrates the connections and exchanges between these managers using state charts, class diagrams, and sequence diagrams from the Unified Modeling Language (UML).It suggests that any embedded device should have PM implemented at the operating system level. Additionally, it suggests interfaces for communication between the Power Manager and the application power management and the device power manager. This object- oriented approach makes it simple for device drivers and application developers to create embedded systems that are faster, easier to maintain, and consume less power.

B. Efficient Coding Practices

Using fixed-point operations instead of floating-point (if precision requirements allow) can lower energy consumption. Efficient coding practices for hardware optimization in embedded systems include: choosing appropriate data types, minimizing unnecessary operations, utilizing hardware-specific features like DMA, employing optimized algorithms, hand-coding critical sections in assembly, utilizing compiler optimizations, and carefully managing memory usage to maximize performance while respecting the constraints of the target hardware.

C. Event-Driven Programming

Event-driven systems only wake up the CPU or peripherals when triggered by an event, reducing unnecessary polling. Example: Sleep modes in wireless sensor nodes until external sensors trigger an interrupt. event-driven programming is a programming paradigm that optimizes hardware utilization by only executing code when a specific event occurs, like a sensor reading change or a hardware interrupt, rather than continuously polling for updates, leading to reduced CPU usage and improved power efficiency; this makes it particularly valuable for resource- constrained embedded systems where minimizing active processing time is crucial.

D. Power Profiling and Analysis

Accurately capturing the target system's power usage trends is necessary for power management. A variety of power profiling methods are available, including software instrumentation tools, monitoring-based power consumption analysis, direct online power consumption measurement, simulations, analytical modeling, and power behavior sampling [10] like the Energy Trace (TI) or ARM’s Kiel MDK provide insights into power-hungry code sections, enabling developers to optimize for energy efficiency. Power profiling" in hardware optimization within embedded systems refers to the process of meticulously measuring and analyzing the power consumption of different components and functionalities within a system to identify areas where power can be reduced, allowing for more efficient hardware design and optimization, particularly in battery-powered devices where power efficiency is critical.

3. System-Level Design

A. Clock Frequency Management

In embedded system design, "clock frequency management" refers to the practice of dynamically adjusting the operating clock frequency of a processor based on the current workload, primarily to optimize power consumption by lowering the clock speed when processing demands are low, and increasing it when high performance is needed - a technique commonly called "Dynamic Voltage and Frequency Scaling (DVFS)". Operating the CPU at lower frequencies when full performance is unnecessary saves energy. Dynamic Frequency Scaling can further reduce power by scaling down the clock when idle.

B. Parallelism

Parallelism refers to the technique of executing multiple tasks or operations seemingly simultaneously by leveraging the capabilities of the hardware, like multi-core processors, to improve system responsiveness and throughput by dividing workload across different processing units, often achieved through techniques like interrupt handling, thread management, and state machines, depending on the specific system requirements. Systems designed to handle tasks in parallel can achieve faster execution and spend more time in a low-power state.

C. Memory Access Optimization

Memory access optimization in embedded systems at the software level design involves techniques to minimize the number of memory accesses required by the program, often by strategically structuring data, utilizing efficient data types, and carefully managing memory allocation to reduce overall memory usage and improve system performance, especially on devices with limited memory capacity. Reducing memory access latency through caching or optimizing data flow can lead to significant energy savings.

IV. ENERGY HARVESTING TECHNOLOGIES

Energy harvesting technologies allow systems to generate their own power from environmental sources. This enables self-sustaining embedded systems with minimal reliance on external power sources.

A. Solar Energy Harvesting

Solar energy harvesting refers to the process of capturing sunlight using photovoltaic (PV) cells to generate electricity that can power small electronic circuits like sensors or microcontrollers, essentially providing a renewable energy source for the system without needing traditional batteries that require frequent replacement. This is often considered the most viable option due to its relatively high power density compared to other energy harvesting methods. Photovoltaic cells Converts sunlight into electricity. They are used in outdoor IoT devices, remote sensors, and solar-powered wearables. Its Challenges include its efficiency decreases in low-light or indoor conditions, requiring efficient power management units (PMUs) to boost and regulate power.

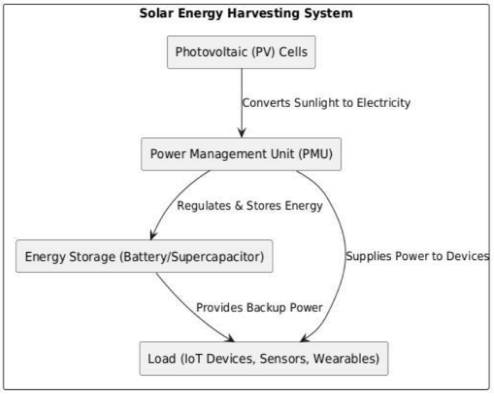

Fig. 2. Solar Energy Harvesting System

B. Thermal Energy Harvesting

- *Thermal Generators(TEGs)*

TEGs use the Seebeck effect to convert temperature differences into electrical energy. Allowing for self-powered operation of small electronic devices by utilizing temperature differences within the system, without any moving parts, making it ideal for compact and reliable energy harvesting in embedded applications. The growing need for recent years have seen a major increase in research into energy collecting devices due to the use of alternative energy sources in low- power electronics. Among other energy sources, thermoelectric materials that make use of the Seebeck effect can transform thermal energy into electrical power [11]. Applications include Industrial settings or body-worn devices that exploit human body heat to generate small amounts of energy. Its challenges are limited power output and dependence on temperature gradients. Design considerations

include Heat source selection by choosing the most suitable heat source with a significant temperature difference.

Thermal interface materials are Using high-conductivity materials to maximize heat transfer to the TEG. Circuit optimization is done by matching the load impedance to the TEG output impedance for maximum power delivery.

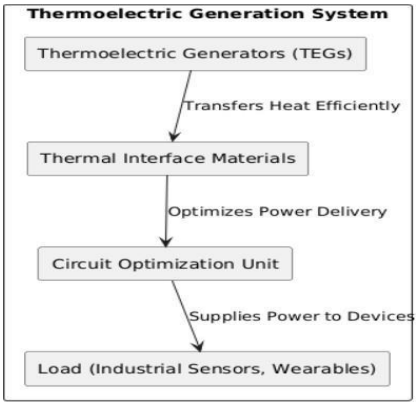

Fig. 3. Thermal Generation System

C. Vibration and Mechanical Energy

i. Piezoelectric Nano generators

A piezoelectric Nano generator (PENG) is a device that harnesses mechanical energy from vibrations and converts it into electrical energy on a nanoscale, making it suitable for powering small, embedded systems by capturing ambient vibrations from the environment, potentially eliminating the need for traditional batteries in certain applications. This technology is particularly promising for low-power electronics due to s ability to harvest energy from subtle movements like footsteps, airflow, or machine vibrations. Mechanical stress is applied to piezoelectric materials, generating electrical energy. Its Applications include Tire pressure sensors, wearables, and industrial equipment monitoring.

ii. Triboelectric Nanogenerators

A triboelectric nano-generator (TENG) is a device that convert mechanical energy, like vibrations, into electrical energy by utilizing the triboelectric effect, making it a promising technology for powering small electronics within embedded systems where ambient vibrations are present, offering a self-powered solution for sensors and other devices without the need for external batteries. Essentially, when two different materials come into contact and slide against each other, they generate electric charges that can be harvested as electricity. Utilizes the contact and separation of two different materials to generate power through the triboelectric effect. Its applications includes Low-power sensors and portable electronics.

iii. Kinetic Energy Harvesters

A kinetic energy harvester, in the context of embedded systems, is a device that converts ambient mechanical vibrations or movement (kinetic energy) into usable electrical energy, often used to power small electronic components within an embedded system, eliminating the need for batteries in certain applications by harnessing energy from the surrounding environment. This is achieved through mechanisms like piezoelectric transducers, electromagnetic coils, or electrostatic capacitors that convert the mechanical vibrations into electrical signals. Convert motion into electrical energy, often used in wearable devices (e.g., watches with kinetic energy harvesting mechanisms).

D. RF Energy Harvesting

The process of capturing radio frequency (RF) electromagnetic waves from the environment and turning them into usable electrical energy to power small, embedded devices that may also be harvesting energy from vibrations or other mechanical sources is known as "RF energy harvesting." This approach essentially provides a hybrid energy harvesting strategy to prolong battery life or even do away with the need for batteries entirely. converts ambient radio frequency (RF) waves from Bluetooth, Wi-Fi, and cellular networks into useful energy. Ultra-low-power Internet of Things devices are powered by these. The potential of ambient UHF/RF energy harvesting as a key technology for enabling the Internet of Things (IoT) and smart skin applications[12].

E. Energy Efficiency Methods

1. EREER: Energy Aware Register File and Execution Unit Using Redundancy in GPGPUs

The utilizing computation redundancy within the Execution Unit (EU) to enhance energy efficiency. The approach, termed EREER (Energy-Aware Register File and Execution Unit by Exploiting Redundancy), is designed to reduce power consumption in GPUs. A data redundancy

exploitation unit identifies redundant data in the input channel, which is a major contributor to GPU power usage. A compression unit at the Register File (RF) input stage then compresses these redundant lanes. The bank arbiter subsequently stores and manages these compressed lanes within the RF. When data is required from the RF, the bank arbiter retrieves the index, decompresses the stored lane, and transmits it to the operand collector. Additionally, the computation reuse unit detects and eliminates redundant tasks, allowing the Stream Processors to be powered down accordingly.

2. *ApproxIt: An Approximate Computing Framework for Iterative Methods*

Approximate computing is a novel design paradigm that leverages the error resilience of various applications, including digital signal processing and multimedia processing, to optimize the trade-off between computational accuracy and power efficiency. ApproxIt, presents an adaptive approximate computing methodology tailored for iterative algorithms[3]. The proposed ApproxIt framework dynamically adjusts approximation modes to reduce computational workload, thereby enhancing energy efficiency without significantly compromising output quality. The ApproxIt framework operates in two distinct phases: 1.Offline Characterization Phase 2.Online Reconfiguration Phase Each application exhibits a unique error resilience profile. The offline phase aims to identify error-resilient and error-sensitive components of the application while evaluating the impact of approximate hardware configurations. In the online phase, ApproxIt dynamically adjusts approximation modes based on real- time computational demands, aligning with the error tolerance levels of the application.

V. POWER OPTIMIZATION TECHNIQUES

These techniques are used to reduce dynamic and static power consumption in embedded systems.

A) Artificial Intelligence-Based Power Optimization:

Traditional power optimization methods in embedded systems, such as Dynamic Voltage and Frequency Scaling (DVFS) and low-power design strategies, have been widely used. AI-driven techniques enhance these approaches by dynamically adjusting system parameters based on real-time data and predictive analytics. In adaptive solo tape and frequency scaling—an extension of DVFS—voltage levels are adjusted in discrete steps according to predefined frequency-based voltage tables. To ensure safe power transfer, large safety margins are incorporated into open- loop systems. Alternatively, Adaptive Voltage Scaling (AVF5) employs specialized analog circuitry to monitor temperature, pressure, and infrared drop, providing continuous feedback for closed-loop voltage scaling. Although this increases system complexity, it results in greater power efficiency over time. The multi-supply voltage (MSV) and multi-mode/multi-corner (MMMC) approaches are combined to achieve DVFS implementation, ensuring that power domains are well-defined across system components. One way to implement DVES designs is by using powerdoises. Furthermore, these definitions of power domains need to match the definitions of power domains on the front end [1].

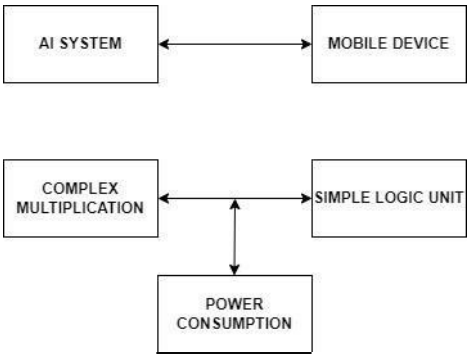

Fig. 4. Artificial Intelligence based power optimization

B) Workload-Dependent Voltage Scaling (WDVS)

Workload-Dependent Voltage Scaling (WDVS) seeks to balance power efficiency and performance by dynamically adjusting voltage levels based on logic path feasibility. It optimizes critical paths and reduces overall power consumption across the chip. This method involves increasing dynamic timing slack (DTS) and subsequently reducing operating voltage until timing constraints are breached, thereby identifying the optimal voltage-performance trade-off. The primary objective of this research is to minimize power consumption in ultra-low- power (ULP) devices, particularly those used in IoT applications. The importance of low power consumption in embedded systems is covered in the study, with a special emphasis on cache memory design. It presents a predictive placement approach designed to lower power consumption in data and

instruction caches. In comparison to traditional set associative cache methods, the technique exhibits significant energy savings, with 71.6 percentage for data cache and 64 percentage for instruction cache, as demonstrated by studies conducted using MiBench embedded benchmarks [2].

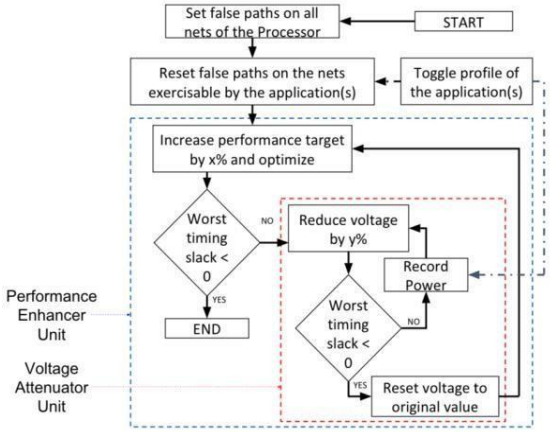

Fig.5. Algorithm of Wide Dynamic Voltage Scaling

C) Dynamic Voltage and Frequency Scaling (DVFS)

The voltage and frequency are scaled based on real-time performance needs. If a system needs less processing power, the frequency and voltage are reduced, resulting in energy savings. Example: Modern processors often have built-in DVFS mechanisms to automatically reduce power when idle or lightly loaded. Significant reduction in dynamic power (dynamic power is proportional to the square of the supply voltage). DVFS-aware is introduced in most modern processors to decrease the power consumption by dynamically adjusting the voltage and frequency at runtime [8]. DVFS dynamically regulates both voltage and frequency to effectively minimize power consumption. By lowering the voltage, it significantly reduces dynamic power, which follows a quadratic relationship with voltage, and also decreases leakage power, which grows exponentially with voltage. These optimizations are crucial for energy-efficient operation, especially in systems with fluctuating workloads. To enhance the accuracy of these adjustments, PVT sensors continuously monitor process variations, voltage levels, and temperature conditions in real time. Their integration plays a vital role in ensuring the efficiency and reliability of DVFS, making them indispensable in modern semiconductor technologies.

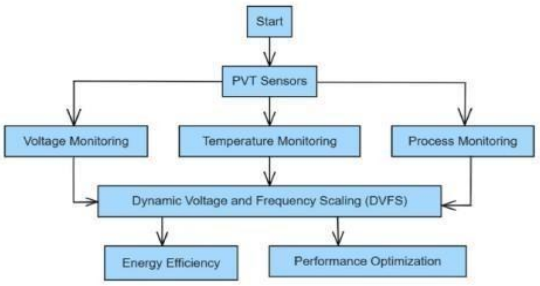

Fig 3. Dynamic Voltage & Frequency Scaling

D. Power Gating

Completely powering down unused components or modules by cutting off the power supply. Power switches are used to disconnect the supply voltage from inactive circuit blocks. Its Applications include Systems with multiple functional blocks (e.g., audio, Wi-Fi) that can be turned off when not in use. Its advantages are Minimizes leakage power in idle states, Suitable for systems with long inactive periods. Its challenges include adding the design complexity, as power gating requires careful management of on/off transitions to avoid errors.

E. Clock Gating

Disabling the clock signal to inactive modules to stop unnecessary switching and reduce dynamic power consumption. This works as in when a module is idle, clock signals are blocked at the input to the flip-flops and registers, preventing unnecessary state changes. They are frequently used in digital signal processing (DSP) and communication systems to reduce power during idle states. This technique Reduces dynamic power without impacting the system state. Easier to implement compared to power gating. The clock signal consumes large amount of the total power; it can be reducing by clock gating i.e. removing the clock signal when it is not required [9]

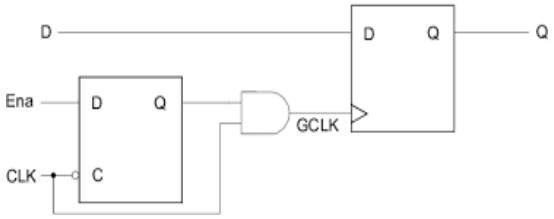

Fig. 5. Clock Gating Circuit

F. Low-Voltage Operation

Low-voltage operation" in power optimization techniques for embedded systems refers to deliberately running the system at a lower voltage

than its maximum rated voltage, which significantly reduces power consumption while still maintaining the necessary functionality, often achieved through techniques like Dynamic Voltage and Frequency Scaling (DVFS). Operating components at the lowest possible voltage for a given frequency can save significant power. Many MCUs allow programmable voltage levels for different operating modes to conserve energy.

G. PSoC

The PSoC family offers a dependable and versatile CPU subsystem that includes SRAM, EEPROM, and Flash memory. It also provides essential system resources such as an oscillator, watchdog timer, sleep timers, along with clock management. Power efficiency in PSoC devices is achieved by carefully regulating analog and digital components. The low-power features are designed to preserve both functionality and performance. Different power modes are defined based on factors like wake-up time, wake-up sources, and power consumption. Moreover, PSoC ensures ultra-low power usage, with its power modes structured in a way that prevents unnecessary energy loss [1].

VI. CONCLUSION

The optimization of power consumption in embedded systems are crucial for enhancing energy efficiency, prolonging battery life, and supporting sustainable technology. This report has examined various methodologies, including low-power hardware design, energy-efficient software techniques, and system-level optimizations.

Dynamic Voltage and Frequency Scaling (DVFS), power gating, and clock gating are important developments in low- power design techniques that allow embedded systems to function with little energy consumption. Furthermore, alternate power sources for self-sustaining embedded systems are offered by energy harvesting technologies such as solar, thermal, radiofrequency, and mechanical energy harvesting.

Both dynamic and static power consumption can be decreased by implementing power optimization techniques such as memory access optimization, event-driven programming, and the usage of low-leakage components. The adoption of Programmable System on Chip (PSoC) further enhances power efficiency by integrating configurable analog and digital components.

As the demand for energy-efficient embedded systems continues to grow, these power-saving techniques will play a pivotal role in shaping the future of IoT, mobile devices, and industrial applications. Ongoing research and innovation will be essential to further refine these strategies, ensuring that embedded systems remain sustainable, efficient, and capable of meeting the yielding needs of modern technology.

VII. REFERENCES

[1] Singha, S., Singh, A. S., Alam, A., & Prasad, S. (2019). "A Study on Power Optimization Techniques in PSoC".0225–0229. https://doi.org/10.1109/iccsp.2019.8698070

[2] Zuo, X. 2011, August 1. Low power performance achievement in embedded system. https://doi.org/10.1109/aimsec.2011.6010609

[3] Segment-aware Energy-efficient Management of Heterogeneous Memory System for Ultra-Low-Power IoT Devices.

[4] T, R., P B, Y., & R, A. K. (2025). Minimization of Power Using the Power Gating Technique to Design a Twisted Ring Counter. 760–763. https://doi.org/10.1109/icmsci62561.2025.10894304.

[5] Indumathi, G., & Ramakrishnan, K. (2011). Study and analysis of power optimization techniques for embedded systems. *IEEE Computer Society Annual Symposium on VLSI*, 365–366. https://doi.org/10.1109/isvlsi.2011.53

[6] Mittal, S. (2014)."A Survey of Techniques For Improving Energy Efficiency in Embedded Computing Systems".cornell university. https://doi.org/10.48550/arxiv.1401.0765

[7] Mixed Signal Microcontroller Data Sheet" (SLAS578D), Texas Instruments Inc., 2007.

[8] Rahimi, P., Wang, X., Prakash, A., & Singh, A. K. (2021). Trends and Challenges in Ensuring Security for Low-Power and High-Performance Embedded SoCs, 226–233. https://doi.org/10.1109/mcsoc51149.2021.00041

[9] S.Panda, S. Sharma, and A. R. Asati, "Clock gating analysis of TG based D flip-flop for different technology nodes," in 2020 IEEE 7th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON), pp. 1–6,

2020.

[10] Y. Liu and H. Zhu., "A survey of the research on Power Management Techniques for High Performance Systems," Software : Practice and Experience., Volume 40., Issue 11., October 2010

[11] Till Huesgen, Peter Woias and Norbert Kockmann, "Design and fabrication of MEMS thermoelectric generators with high temperature efficiency", *Sensors and Actuators A: Physical*, vol. 145–146, pp. 423-429, 2007.

[12] Kim, S., Georgiadis, A., Tentzeris, M. M., Niotaki, K., Vyas, R., Bito, J., & Collado, A. (2014). Ambient RF Energy-Harvesting Technologies for Self-Sustainable Standalone Wireless Sensor Platforms. Proceedings of the IEEE, 102(11), 1649–1666. https://doi.org/10.1109/jproc.2014.2357031