The American Journal of Engineering and Technology
ISSN 2689-0984 | Open Access

TYPE Original Research
PAGE NO. 32-40

Check for updates

Power And Verification Challenges In Ultra-Low-Power Integrated Circuits: A Critical Review Of Design And Automation Techniques

Rizky Pratama
Faculty of Electrical Engineering and Information Technology, Sepuluh Nopember Institute of Technology, Surabaya, Indonesia

Kwame Nkrumah
Department of Computer Engineering, Kwame Nkrumah University of Science and Technology, Kumasi, Ghana

Yuri Ivanovich Petrov
Faculty of Computer Science and Technology, Saint Petersburg State University, Saint Petersburg, Russia

OPEN ACCESS

SUBMITED 10 September 2025
ACCEPTED 29 October 2025
PUBLISHED 05 November 2025
VOLUME Vol.07 Issue 11 2025

CITATION

Abstract: Purpose: This critical review systematically examines the complex interplay between ultra-low-power (ULP) design methodologies and the corresponding verification challenges in modern integrated circuits. The imperative for pervasive, battery-operated devices (IoT, wearables) has pushed design-for-power techniques to their limits, simultaneously creating significant hurdles for ensuring functional correctness.

Methodology: The paper first establishes the theoretical foundation of power dissipation in CMOS circuits, followed by a systematic survey of leading ULP design techniques, including dynamic voltage/frequency scaling, power gating, and multi-threshold CMOS. It then evaluates state-of-the-art power estimation and, crucially, formal and simulation-based methodologies necessary for verifying the functional integrity and power intent, as formalized by the Unified Power Format (UPF).

Findings: Aggressive power reduction techniques, particularly power gating, fundamentally alter the circuit's state and timing characteristics, rendering traditional verification flows insufficient. Formal verification, specifically equivalence checking and property checking based on Satisfiability (SAT) and Binary Decision Diagrams (BDD), is increasingly indispensable for exhaustively validating power

management logic and state retention mechanisms.

Originality: This review offers a holistic synthesis, bridging the gap between ULP design methodology and its formal verification requirements, providing a foundational resource for researchers and practitioners navigating the dual constraints of energy efficiency and functional integrity in next-generation VLSI.

Keywords: Ultra-Low-Power Design, VLSI, Power Gating, Formal Verification, Unified Power Format (UPF), Power Estimation, CMOS.

1. Introduction

1.1. Background and Motivation

The proliferation of Internet of Things (IoT) devices, wearable electronics, and mobile computing platforms has established energy efficiency as a paramount design metric for Very Large-Scale Integration (VLSI) systems. These devices often rely on finite battery resources, making the reduction of power dissipation a primary constraint that frequently outweighs the traditional focus on maximum operating frequency. Ultra-low-power (ULP) design is no longer a niche field but a central challenge in scaling down semiconductor technology. Early foundational work provided the necessary mathematical models and circuit-level understanding that underpins modern low-power techniques.

The fundamental relationship governing dynamic power dissipation shows a quadratic dependence on the supply voltage, alongside a linear dependence on the switching activity and the operating frequency. Consequently, voltage scaling has historically been the most effective strategy for reducing power. However, as supply voltages approach the threshold voltage of transistors in advanced process nodes, static power—primarily due to leakage current—begins to dominate the overall power budget. This shift necessitates the implementation of more aggressive and complex power management schemes that target both dynamic and static components.

1.2. The Dual Imperative: Power and Correctness

The deployment of sophisticated power management techniques, while essential for energy saving, introduces significant complexity into the design and verification flow. Techniques such as power gating, which selectively shuts down inactive circuit blocks, require careful orchestration of power switches, isolation cells, and state retention mechanisms. A failure in the control logic for these elements is associated with functional errors, data corruption, or catastrophic design failure.

To manage this complexity, industry adopted the Unified Power Format (UPF), codified in the IEEE Standard 1801, as a critical tool. UPF allows designers to formally capture the intended power architecture—specifying power domains, power switches, isolation, and retention—separate from the Register Transfer Level (RTL) functional description. While UPF standardizes the power intent, the functional verification of the implementation of this intent remains a formidable challenge. Traditional simulation-based verification struggles to achieve the necessary coverage for the myriad of complex power-up, power-down, and retention scenarios. The dual imperative, therefore, is to achieve maximal power efficiency without compromising functional correctness.

1.3. Literature Review and Research Gaps

The academic and industrial literature surrounding ULP design is rich and spans several decades. Foundational textbooks thoroughly detail the physics of power dissipation and introduce core techniques like clock gating and voltage scaling. Further research has explored the efficacy of dynamic power management, presenting design techniques and computer-aided design (CAD) tools necessary for their implementation. The literature also provides comprehensive surveys on power estimation techniques, covering models from behavioral to transistor level, allowing designers to predict power consumption throughout the design hierarchy. Separately, the field of formal hardware verification provides methodologies for rigorously proving the correctness of digital circuits, utilizing techniques such as equivalence checking and model checking.

Despite this extensive background, two significant research gaps persist in the current body of knowledge:

Gap 1: Insufficient Holistic Treatment of Co-Design and Co-Verification: Current reviews often treat low-power design and verification as distinct disciplines. There is a demonstrable need for a holistic synthesis that systematically connects a specific ULP design choice (e.g., multi-voltage islands) directly to the necessary verification methodology (e.g., formal domain crossing checks) required to validate its correctness.

Gap 2: Need for Detailed Comparative Analysis in Verification: While the importance of formal methods in

verification is recognized, a detailed, in-depth comparative analysis of the practicality, computational limits, and application domains of formal verification (e.g., SAT-based) versus advanced simulation-based verification (e.g., constrained random testing with assertions) for complex power management schemes remains a critical requirement. This gap is particularly evident when considering the scalability of these techniques to designs exceeding billions of transistors.

1.4. Contribution and Paper Structure

This paper addresses these gaps by providing a structured, integrated analysis of ULP design and verification. The structure is as follows: Section 2 details the theoretical framework for power minimization and surveys the key architectural and circuit-level design techniques. Section 3 presents the core verification methodologies, focusing on the role of UPF and the application of both simulation and formal methods. Section 4 provides a comprehensive discussion, synthesizing the design and verification trends and outlining future research directions.

2. METHODS (Theoretical Framework and Design Techniques)

2.1. Theoretical Underpinnings of Low-Power Design

Power dissipation in CMOS circuits is traditionally partitioned into three main components: dynamic power, short-circuit power, and static power.

Dynamic Power P^{dyn}: This component is consumed during the charging and discharging of load capacitances C^L when a logic transition occurs. It is expressed as:

$$P^{dyn} = \alpha C^L V^{dd2} f$$

where α is the switching activity factor, C^L is the total load capacitance, $V_(DD)$ is the supply voltage, and f is the clock frequency. The quadratic dependence on $V_(DD)$ makes voltage scaling the most potent technique for dynamic power reduction.

Short-Circuit Power P_{sc} This power is dissipated when

both the NMOS and PMOS transistors in a CMOS gate are momentarily "on" during a voltage transition, creating a direct path from $V_(DD)$ to ground. While usually a smaller fraction of the total power, accurate simulation remains necessary to account for its contribution.

Leakage current, which includes subthreshold, gate oxide, and junction leakage, has become increasingly significant as feature sizes shrink and threshold voltages Vtare reduced to maintain performance. Minimizing $V_(DD)$ to reduce dynamic power often requires a proportional reduction in Vt to meet timing constraints, which exacerbates the static power problem exponentially.

2.2. Architectural and Circuit-Level Power Reduction

The shift in the power profile has led to a suite of highly integrated power-reduction techniques.

2.2.1. Voltage and Frequency Scaling (DVFS and DVS)

Dynamic Voltage and Frequency Scaling (DVFS) is an established technique where the operating voltage and frequency are adjusted dynamically based on the computational load required by the application. When the workload is light, the voltage and frequency are simultaneously lowered, leading to a significant reduction in dynamic power. A variation, Dynamic Voltage Scaling (DVS), focuses primarily on voltage adjustment while frequency is implicitly limited. The effectiveness of DVFS is highly dependent on the design of the on-chip power management unit (PMU), which must rapidly and stably transition between voltage levels.

2.2.2. Power Gating and State Retention

Power Gating is the most effective technique for reducing static (leakage) power in inactive circuit blocks. It involves inserting a sleep transistor (header or footer switch) in series with the power rail of the block. When the block is inactive, the switch is turned off, effectively isolating the block from the supply and drastically reducing leakage current.

However, power gating necessitates careful handling of the block's internal state. When the block is powered down, its internal register values are lost. State retention flip-flops (SRFFs) are employed to preserve critical state information during the power-down sequence, which is then restored upon power-up. Furthermore, isolation cells are required at the boundary of the power-gated block to prevent the propagation of floating output signals (which could cause crowbar currents in downstream active logic) to adjacent active power domains. The complex sequencing of the retention, isolation, power-down, and power-up phases is a primary source of verification challenge.

2.2.3. Multi-Voltage and Multi-Threshold Design

Multi-Voltage (MV) Design utilizes different voltage domains for different functional blocks, allowing non-critical paths to operate at a lower V_{DD} than performance-critical paths. This strategy significantly reduces power while maintaining overall system throughput. However, MV designs require level shifters at the interfaces between domains operating at different voltages to prevent reliability issues and ensure correct signal transmission.

Multi-Threshold (MTCMOS) Design is a static power reduction technique that utilizes transistors with different threshold voltages Vt within the same circuit. High-Vt transistors, which have lower leakage but are slower, are used in non-critical paths or for the power-gating switch itself. Low-Vt transistors, which are faster but leak more, are reserved for performance-critical paths. The integration of various transistor types requires sophisticated design and synthesis tools.

2.3. Power Estimation and Modeling Techniques

Accurate prediction of power consumption is essential for guiding design trade-offs. Techniques span a hierarchy of abstraction:

2.3.1. Behavioral and RTL-Level Estimation

At the highest level, RTL-level estimation utilizes switching activity extracted from functional simulation and approximate models for circuit capacitance. This early estimation is fast and instrumental for high-level architectural decisions, allowing designers to evaluate the power implications of different micro-architectures before committing to a detailed gate-level implementation. Behavioral-level techniques extend this, estimating power based on the functional description and a deeper understanding of the system's operational modes, though accuracy is inherently limited by the high abstraction level.

2.3.2. Gate-Level and Transistor-Level Simulation

Gate-level power simulation provides a higher degree of accuracy by using library characterization data, which includes precise capacitance and power models for standard cells, alongside detailed net switching activity. This level is essential for final design sign-off. Transistor-level simulation (e.g., using SPICE) provides the highest accuracy, modeling device physics and actual current flow. However, due to its extreme computational cost, this is typically limited to small, critical circuit blocks. The challenge in power estimation lies in finding the optimal balance between computational efficiency and the required level of accuracy across the entire design hierarchy.

3. RESULTS (Verification Methodologies and Case Studies)

3.1. Verification of Power Intent using UPF

The complexity introduced by the design techniques discussed in Section 2 necessitates a formal, standardized way to communicate the power architecture. The Unified Power Format (UPF) is a standard specification language used to define the power domains, the isolation strategy, the retention strategy, and the operational sequences of power management features.

3.1.1. The Role of IEEE Standard 1801

The UPF standard ensures that the intended power strategy is correctly interpreted and implemented by downstream electronic design automation (EDA) tools—from synthesis and place-and-route to formal verification and simulation. The UPF file acts as the single source of truth for power intent. It defines power domains (e.g., specifying which blocks share a supply), the placement of power switches, the location and type of isolation cells, and the logic for controlling the power sequences. Crucially, the standard also supports the definition of abstract power states and the legal state transitions, which become the fundamental properties to be verified.

3.1.2. Challenges in UPF-Aware Simulation

Traditional simulation-based verification (dynamic verification) remains the workhorse for functional validation. However, for power-aware designs, the simulation environment must become UPF-aware. This involves correctly modeling the behavior of power switches (e.g., high impedance output when off), isolation cells (e.g., fixing outputs to a safe value), and state retention logic. The challenge is combinatorial: verifying the core functionality across all required power state transitions (e.g., idle-to-sleep, sleep-to-active, active-to-deep-sleep) and under various timing conditions related to the wake-up and power-down handshake protocols. Achieving high coverage for these transient power management scenarios through simulation alone is extremely difficult and time-consuming.

3.2. Formal Verification in Power-Aware Design

Formal verification offers a complementary approach to

simulation by mathematically proving or disproving the correctness of the design relative to a specification, offering exhaustive coverage. Its role is becoming indispensable for low-power designs where transient states are complex and mission-critical.

3.2.1. Equivalence Checking for Low-Power Synthesis

Low-power synthesis tools often perform complex optimizations, such as inserting clock-gating logic, replacing standard cells with multi-threshold versions, or even modifying the functional description to insert power-gating switches and state retention logic. Sequential equivalence checking (SEC) is used to formally prove that the netlist resulting from the power-aware synthesis is functionally equivalent to the original RTL, ensuring that the power optimizations have not inadvertently introduced functional bugs. This process relies heavily on underlying technologies like Boolean Satisfiability (SAT) solvers and Binary Decision Diagrams (BDDs) to manage the complexity of comparing two circuit descriptions.

3.2.2. Formal Methods for Power Gating Correctness

The most critical application of formal methods is validating the power management logic itself. This involves two main areas:

Isolation and Retention Correctness: Formal property checking is used to prove that when a block is powered down, all its outputs are correctly driven by isolation cells (preventing floating nodes) and that critical internal state is properly captured by the state retention flip-flops before the power switch is opened.

Power Switch Control Logic: The sequence and timing of signals controlling the power switch, isolation cells, and retention registers must be formally proven to operate correctly under all specified conditions. For example, the formal proof must ensure that the power switch is never activated (turned off) before all necessary data has been saved to the retention registers.

3.3. Case Study: Verification Flow for a Power-Gated Block (Expanded)

Power gating, a technique that leverages sleep transistors to virtually eliminate leakage power in idle blocks, is conceptually simple but practically complex. This complexity stems from the intricate sequencing required to manage the block's power state, isolate its outputs, and retain its critical data. The verification flow must therefore be multi-layered, culminating in the rigorous application of formal property checking.

3.3.1. Detailed Sequencing and Control Logic

The power-gating control logic is a small but safety-critical state machine that orchestrates four key events:

Retention Save: Critical state data from the functional registers is transferred into the low-leakage state retention flip-flops (SRFFs). This must occur before the power switch is opened.

Power-Down/Sleep: The control signal to the sleep transistor (the sleep signal) is asserted, opening the power switch and isolating the block's core from the virtual power rail.

Isolation Enable: Isolation cells, placed at the block's outputs, are activated to clamp the downstream signals to a known, safe value (usually 0 or 1), preventing high-impedance (X) propagation. This should ideally occur concurrent with or immediately preceding the power-down sequence.

Power-Up/Restore: The sleep signal is de-asserted to close the power switch. Once the virtual power rail has stabilized (checked via a power-good signal), the data stored in the SRFFs is restored to the functional registers.

The correctness of this entire process hinges on the absence of race conditions or out-of-sequence operations. For example, if the power switch opens even momentarily before the retention data is saved, the state is corrupted. If isolation cells are not activated correctly, the functional integrity of downstream logic is jeopardized.

3.3.2. Formal Property Specification for Power Gating

Formal verification, through model checking, requires the specification of design properties in a formal language (e.g., Linear Temporal Logic or SVA). These properties are then mathematically proven against the design's state machine.

The Core Properties for Power-Gated Blocks:

P1: Safety Property (Isolation): Globally, if the power-switch-open control signal is asserted, then all block output signals must be driven by an active isolation cell.

This is specified as a safety property, ensuring that an undesirable state (floating output) is never reached. Formal tools search the entire state space to find a counterexample where the switch is open and an isolation cell is inactive.

P2: Liveness Property (Restoration): Globally, it must be

the case that if a power-down sequence is initiated and successfully completed, then the subsequent power-up sequence will eventually lead to the correct restoration of the saved state.

This property is more challenging as it involves time and causality. It requires formal modeling of the power-up sequence and the power-good signal stabilization time.

P3: Sequential Dependency Property (Retention): For any transition where the virtual power rail is powered down, it must be true that the retention-save control signal was asserted at least one clock cycle prior to the assertion of the sleep signal.

This is the most critical check for data integrity. The formal tool models the finite state machine of the power control logic and proves that the transition from 'Active' to 'Saving' to 'Sleep' is strictly ordered and non-violable.

3.3.3. The Role of SAT-Based Verification Engines

The engine underpinning most modern formal verification tools is the Boolean Satisfiability (SAT) solver. For a typical digital circuit, the verification problem is transformed into a SAT problem.

Transformation Process:

Bounded Model Checking (BMC): The property check (e.g., P3) is unrolled for a finite number of time steps (k). The entire unrolled circuit, including the property assertion (which is negated, as the tool searches for a counterexample), is converted into a single massive Boolean formula.

Conjunctive Normal Form (CNF): The Boolean formula is then converted into Conjunctive Normal Form (CNF), a format consisting of a conjunction ($\land$) of clauses, where each clause is a disjunction ($\lor$) of literals.

SAT Solver Execution: A highly optimized SAT solver (e.g., based on the DPLL algorithm or modern Conflict-Driven Clause Learning (CDCL) techniques) attempts to find a variable assignment that makes the CNF formula true.

If a satisfying assignment is found, it represents a counterexample—a sequence of inputs and initial states that violates the specified property (i.e., a bug exists in the design). The sequence of assignments provides the complete debug trace.

If the solver proves the formula is unsatisfiable (UNSAT), it means that no counterexample exists within the bound k, suggesting the property holds.

Addressing Scalability with Inductive Proof:

A major limitation of BMC is the depth k. Proving a property holds for a limited time is insufficient for guarantees. This is where Inductive Property Checking is utilized. This method attempts to prove a property using the principle of mathematical induction, which involves three steps:

Base Case (Initialization): Prove the property holds in the initial reset state. (Similar to BMC with $k=1$).

Inductive Step (Propagation): Prove that if the property holds at any arbitrary time t, it must also hold at time $t+1$. This involves translating the system transition relation into a SAT problem.

Safety Check (Inductive Hypothesis): This step ensures that the set of states where the property holds is reachable from the initial state, often implicitly covered by the first two steps for simple designs.

If both the base and inductive steps are proven UNAT by the SAT solver, the property is considered proven for all time steps, providing the necessary rigorous assurance for the power-gating control logic. The use of sophisticated SAT solvers, particularly those with incremental solving capabilities and conflict-driven learning, is critical to managing the complexity associated with verifying state retention and sequencing logic, which inherently involves a large number of state variables.

3.3.4. Limitations of Formal Verification in ULP Design

Despite its power, formal verification has practical limitations:

State-Space Explosion: For very large designs or blocks with deeply sequential logic (e.g., large FIFO controllers or complex memory access units), the number of reachable states can become mathematically intractable. The size of the CNF formula can exceed the memory and time capacity of the SAT solver. The entire chip's power management must often be partitioned into smaller, verifiable units.

Property Completeness: Formal verification only proves that the design meets the properties that are explicitly written. If a critical corner-case property is missed, the design flaw will not be found formally. This necessitates expert knowledge to ensure the property set is complete relative to the power intent.

Abstraction Fidelity: Formal tools operate on a model of the design (usually the gate-level netlist). They do not

natively model deep analog effects, such as the exact dynamic characteristics of the $V_{virtual}$ power rail during recovery or the precise timing of noise injection. The reliance on discrete clock cycles and ideal switch behavior introduces an abstraction gap that must be bridged by additional Spice/analog simulation.

4. DISCUSSION

4.1. Synthesis of Power Design and Verification Trends

The journey towards achieving minimal energy consumption in integrated circuits is fundamentally intertwined with the challenge of ensuring functional integrity. The most effective power reduction techniques, such as aggressive voltage scaling and dynamic power gating, are also the ones that introduce the most significant verification hurdles. Reducing the supply voltage pushes the circuit closer to its noise margin, increasing sensitivity to variations. Implementing power gating introduces non-functional, temporal aspects—the complex sequences of power-up and power-down—that must be proven correct.

The widespread adoption of the Unified Power Format (UPF) has been instrumental in bridging the design and verification domains, providing a common language for power intent. However, UPF's success is contingent upon the robustness of the verification tools that process it. The trend indicates a necessary pivot from purely dynamic (simulation-based) verification towards a hybrid methodology where formal verification is non-negotiable for validating the correctness of the power management control logic. Formal methods excel in proving the absence of bugs in state machine control and sequencing, which is exactly the nature of power-up and power-down protocols.

4.2. Comparative Analysis of Verification Approaches

The choice between dynamic simulation and formal verification is not a dichotomy but a strategic decision based on the design's complexity and the required assurance level.

Feature	**Dynamic Simulation (Constrained Random)**	**Formal Verification (Property Checking/Equivalence)**
Coverage	Statistical; dependent on test generation quality; difficult for deep corner cases.	Exhaustive for the property under check; guaranteed coverage for critical logic.
Complexity	High setup cost (testbench, models); manageable run-time complexity for large designs.	High complexity for writing comprehensive properties; can face state-space explosion.
Application	Core functional validation; performance checking; transient power-up/down sequencing.	**Safety-critical power sequencing; isolation correctness; equivalence checking after synthesis.**
Result	Shows the presence of bugs.	Proves the absence of bugs (for the defined property).

For ULP designs, dynamic simulation is effective for checking high-level power consumption profiles and validating the overall functional flow under various power modes. However, formal verification provides the necessary proof of correctness for critical power infrastructure elements (switches, isolation, retention) whose failure would be catastrophic. The industry is moving towards a flow where the functional core is primarily verified by simulation, while the power intent implementation and its control logic are formally

proven.

4.3. Technological Implications and Future Research Directions

The continuous scaling of semiconductor technology poses perpetual challenges. Technologies like Silicon-on-Insulator (SOI) devices, by minimizing junction capacitance and substrate leakage, directly support ULP objectives. The emergence of FinFET and Gate-All-Around (GAA) architectures, while improving electrostatic control and reducing leakage, also introduce new physical modeling complexities that directly impact the accuracy of power estimation and the required voltage/current modeling in verification.

Future research should focus on:

Co-Design Automation: Developing highly automated CAD tools that can ingest the UPF power intent and automatically synthesize the necessary power management logic (switches, isolation) and concurrently generate the necessary formal properties to verify that implementation.

Scalability of Formal Methods: Addressing the state-space explosion problem in formal verification to allow exhaustive checking of larger, more complex power domains and multi-level power management hierarchies.

Validation of Analog/Digital Interfaces: The intersection of digital power management control and the analog power conversion/regulation circuitry (e.g., DC-DC converters, LDOs) remains a significant challenge. Methods for formally verifying the interaction between the digital control state machine and the analog power delivery network are highly needed.

4.4. Discussion Limitations

The scope of this review, while aiming for a holistic treatment, primarily concentrates on digital CMOS ULP design techniques and their corresponding verification flows. It does not provide an exhaustive treatment of analog or mixed-signal power management, which involves a distinct set of design and verification challenges. Furthermore, the analysis of specific performance metrics for different formal tools (e.g., run-time for various SAT solver configurations) is abstracted, focusing instead on the methodological requirement. Finally, industry practice often includes proprietary, unpublished techniques, meaning the analysis relies on publicly available academic and industrial standards documents, which may not capture the most cutting-edge, internal-use methodologies.

REFERENCES

1. Chandrakasan, A. P., & Brodersen, R. W. Low power digital CMOS design. Kluwer Academic Publishers.
2. Rabaey, J. M. Digital integrated circuits: a design perspective. Prentice Hall.
3. Benini, L., & De Micheli, G. Dynamic power management: design techniques and CAD tools. Kluwer Academic Publishers.
4. Najm, F. N. A survey of power estimation techniques in VLSI circuits. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2(4), 446-455.
5. Kang, S. Accurate simulation of power dissipation in VLSI circuits. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 17(5), 438-444.
6. Nagaraj, V. (2025). Ensuring low-power design verification in semiconductor architectures. Journal of Information Systems Engineering and Management, 10(45s), 703–722. https://doi.org/10.52783/jisem.v10i45s.8903
7. Kropf, T. Introduction to formal hardware verification. Springer.
8. Claessen, K. A survey of proof checking techniques for hardware verification. In Formal Methods for Real-Time and Probabilistic Systems (pp. 243-267). Springer.
9. Mishchenko, A., Brayton, R., & Jiang, S. SAT-based logic optimization and equivalence checking. In Proceedings of the 40th annual Design Automation Conference (pp. 448-453). ACM.
10. Brand, D. Verification of large synthesized designs. In Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (pp. 534-537). IEEE Computer Society Press.
11. IEEE Standard 1801-2015. IEEE Standard for Design and Verification of Low Power Integrated Circuits. IEEE.
12. Cristoloveanu, S., & Li, S. Electrical characterization of silicon-on-insulator materials and devices. Springer.
13. Building Compliance-Driven AI Systems: Navigating IEC 62304 and PCI-DSS Constraints. (2025). International Journal of Networks and Security,

5(01), 62-90. https://doi.org/10.55640/ijns-05-01-06

14. Vikram Singh, 2025, Adaptive Financial Regulation Through Multi-Policy Analysis using Machine Learning Techniques, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 14, Issue 04 (April 2025)