Energy-Efficient Semiconductor Device Design for Low-Power Electronic Computing Applications

Ananya Iyer

India

Abstract

This study presents an energy-efficient semiconductor device design framework for low-power electronic computing applications, focusing on reducing power dissipation while maintaining switching performance, operational reliability, and computational efficiency. The proposed methodology investigates advanced transistor architectures, material characteristics, and circuit-level optimization techniques to minimize dynamic and static power consumption. Device modeling incorporates threshold voltage, leakage current, gate capacitance, switching frequency, and operating voltage to characterize energy dissipation under different computational workloads. Technology Computer-Aided Design simulations are proposed to evaluate electrical characteristics, carrier transport behavior, and short-channel effects across alternative device configurations. Adaptive voltage scaling and leakage reduction strategies are integrated to improve energy efficiency without significantly compromising processing speed. Performance evaluation considers power-delay product, subthreshold leakage, switching energy, thermal stability, and device scalability. The proposed framework supports the development of energy-efficient semiconductor technologies for portable electronics, embedded processors, Internet of Things devices, and next-generation low-power computing systems.

Keywords: Semiconductor Device Design; Low-Power Electronics; Energy Efficiency; Leakage Current Reduction; Power-Delay Optimization.

1. Introduction

Enterprise applications depend on reliable diagnostics to detect runtime errors, failed workflows, API issues, slow transactions, and integration faults [1]. Logs are the main evidence source for understanding what happened before, during, and after an incident. Effective application diagnostics require structured logging, traceability, contextual metadata, error classification, and operational visibility [2]. However, fixed logging strategies often create two problems. Low-detail logs may miss important causes, while high-detail logs may generate excessive storage cost and alert noise [3].

The main challenge is that diagnostic needs change during runtime. A normal transaction may require only basic logs, but a failing workflow, repeated exception, or security-sensitive event may need deeper traces [4]. Adaptive logging improves this process by changing log depth according to system condition, transaction risk, and error behavior [5]. AI-assisted monitoring can identify when additional diagnostic detail is needed and when logging can return to normal levels. This article proposes an adaptive logging framework for enterprise application diagnostics.

2. Methodology

The proposed framework captures log metadata from application servers, API gateways, databases, background jobs, authentication modules, and integration services [6]. It records event type, severity, timestamp, user role, transaction ID, request path, error frequency, response time, affected component, and workflow state. These features are converted into diagnostic context profiles. Adaptive logging requires context because the same event may be low-risk in one workflow but critical in another [7].

The logging controller classifies runtime states as normal, warning, error-prone, performance-risk, security-sensitive, workflow-critical, or incident-active [8]. Based on this classification, it adjusts log level, trace depth, parameter capture, correlation ID tracking, and diagnostic enrichment. Lightweight scoring and policy rules are used so logging decisions remain fast and explainable. Sensitive fields are masked before storage [9], and transaction-critical events are linked with workflow and database identifiers for easier diagnosis.

The framework is evaluated using simulated enterprise applications involving APIs, database workflows, authentication events, scheduled jobs, and service integrations. Static logging is compared with adaptive logging under repeated exceptions, slow requests, failed jobs, API timeout, suspicious login, and workflow interruption scenarios [10]. Evaluation metrics include diagnosis time, log volume reduction, root-cause evidence quality, missed-event rate, storage overhead, alert usefulness, and administrator acceptance rate. Feedback from resolved incidents, ignored logs, and administrator notes is used to refine logging policies over time.

3. Results and Discussion

The results show that adaptive logging improves diagnostics by capturing richer evidence only when runtime risk increases. In the baseline condition, static logs either lack enough detail or produce large volumes of low-value records. With the proposed framework, detailed traces are activated for high-risk events and reduced during normal execution. The strongest improvement appears in intermittent failures, where deeper logs are needed only around the failure window.

The discussion shows that adaptive logging must balance diagnostic depth with privacy and storage cost. Excessive logging can expose sensitive data or overload monitoring systems, while weak logging may hide failure causes. The main limitation is that log-level decisions depend on accurate runtime classification. Regular policy review and masking controls are necessary for safe long-term use.

4. Conclusion

This article presented an adaptive logging framework for enterprise application diagnostics. The framework adjusts log detail using runtime context, risk scoring, workflow state, and error behavior. It improves troubleshooting by increasing diagnostic evidence during abnormal conditions while reducing unnecessary log noise during normal operation. The study shows that adaptive logging can strengthen enterprise diagnostics when combined with structured metadata, privacy controls, and continuous incident feedback.

References

1. Xu, P., Liang, J., Li, H., Liu, F., Tie, J., Jiao, Z., ... & Lu, J. (2020). Device performance limits and negative capacitance of monolayer GeSe and GeTe tunneling field effect transistors. *RSC advances*, *10*(27), 16071-16078.
2. Zhai, Y., Feng, Z., Zhou, Y., & Han, S. T. (2021). Energy-efficient transistors: suppressing the subthreshold swing below the physical limit. *Materials Horizons*, *8*(6), 1601-1617.
3. rao Gondi, S. (2019). Predictive Modeling of Wafer Acceptance Test Parameters for Semiconductor Yield Enhancement. Emerging Digital Systems, 6, 33-38.
4. Chauhan, V., & Samajdar, D. P. (2021). Recent advances in negative capacitance FinFETs for low-power applications: a review. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, *68*(10), 3056-3068.
5. Cai, Y., Zhang, Z., Zhang, Q., Xiang, J., Xu, G., Wu, Z., ... & Yin, H. (2021). Investigation of time domain characteristics of negative capacitance FinFET by pulse-train approaches. *Journal of Semiconductors*, *42*(11), 114101.
6. Yadav, S., Kondekar, P. N., Upadhyay, P., & Awadhiya, B. (2022). Negative capacitance based phase-transition FET for low power applications: Device-circuit co-design. *Microelectronics Journal*, *123*, 105411.
7. Solay, L. R., Kumar, N., Amin, S. I., Kumar, P., & Anand, S. (2022). Design and performance analysis of gate-all-around negative capacitance dopingless nanowire tunnel field effect transistor. *Semiconductor Science and Technology*, *37*(11), 115001.

8. rao Gondi, S. (2021). Big Data Analytics Framework for Process Optimization and Yield Improvement in Semiconductor Production. Journal of Privacy-Aware Computing Systems, 8, 52-57.
9. Aziz, J., Kim, H., Hussain, T., Lee, H., Choi, T., Rehman, S., ... & Kim, D. K. (2022). Power efficient transistors with low subthreshold swing using abrupt switching devices. *Nano Energy*, *95*, 107060.
10. Ravipati, D. P., van Santen, V. M., Salamin, S., Amrouch, H., & Panda, P. R. (2023). Performance and energy studies on NC-FinFET cache-based systems with FN-McPAT. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, *31*(9), 1280-1293.