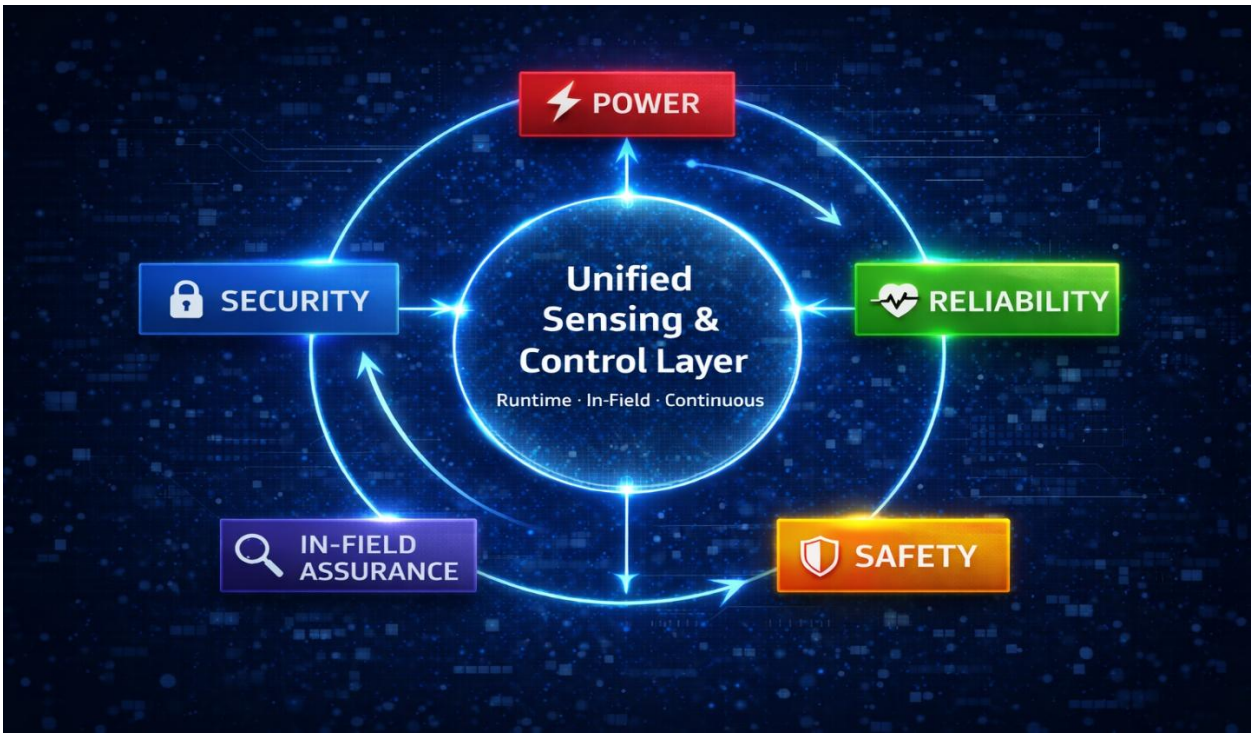
VegaSemi
Smart Silicon



Unified Silicon Runtime™ Architecture

Conceptual Framework

Power Management • Functional Safety • Reliability • Security

White Paper | March 2026

Ben Zaryouh
Founder & CEO
VegaSemi
ben@vegasemi.com

Table of Contents

1 Executive Summary

Modern System-on-Chip (SoC) platforms are operating under converging pressures from advanced technology scaling and emerging application demands. Nanoscale technologies introduce tighter operating margins, greater variability, stronger thermal and aging sensitivity, and increasingly complex fault behavior. At the same time, AI cloud systems, autonomous platforms, robotics, and industrial machines demand higher compute density, better energy efficiency, and stronger guarantees for safety, security, reliability, and in-field integrity.

Conventional runtime architectures are increasingly inadequate under these conditions. Power management, safety, security, reliability, and diagnostics are typically implemented as isolated subsystems with independent sensing, monitoring, and control mechanisms. This fragmented model depends heavily on distributed proxy sensors, which provide indirect visibility and incomplete spatial coverage while increasing area, power, routing, calibration, and integration overhead. As SoCs operate closer to their physical limits, such architectures become less accurate, less efficient, and more difficult to scale.

This paper introduces the Unified Silicon Runtime Architecture (USRA), a new runtime paradigm built on consolidated telemetry, direct observability within the functional execution fabric, and coordinated cross-domain control. Rather than treating runtime functions as separate infrastructures, USRA establishes a shared telemetry and decision framework capable of addressing both fast transient effects and slower evolving silicon conditions through a common execution-aware control model.

Built on shared in-situ telemetry and direct functional observability, USRA enables multiple critical runtime technologies, including timing-aware voltage adaptation for power optimization, continuous margin tracking for reliability and aging management, proactive runtime monitoring for safety and functional integrity, execution-aware in-field diagnostics, and runtime security monitoring for transient attack mitigation and physical-layer anomaly detection.

By consolidating telemetry and coordinating response across traditionally independent domains, USRA improves power efficiency, performance stability, area utilization, safety coverage, security resilience, and long-term reliability while reducing conservative guardbands and duplicated monitoring infrastructure. More fundamentally, it shifts runtime management from a fragmented, domain-specific model to a unified, silicon-aware control framework better suited to the demands of modern SoCs.

2 Introduction

Modern System-on-Chip (SoC) platforms are at a critical juncture, driven by the relentless pace of advanced technology scaling and the escalating demands of AI-driven and Physical AI workloads. These forces require a fundamental re-evaluation of how silicon platforms are managed at runtime. This paper develops the case for a new runtime architecture built around shared observability, consolidated telemetry, and coordinated cross-domain control. The sections that follow examine the pressures and limitations that make such a shift necessary, then introduce the Unified Silicon Runtime Architecture (USRA) and the runtime technology domains it enables.

3 SoC Runtime Architectures: Current State and Key Limitations

Key Observations:

- Technology scaling is increasing runtime uncertainty and reducing physical margin.
- Modern AI-driven systems demand tighter efficiency, correctness, and system integrity.
- Conventional runtime architectures remain fragmented across domains.
- Legacy sensing and monitoring techniques are poorly matched to modern SoC behavior.
- A unified runtime paradigm is needed to enable shared observability and coordinated control.

Modern SoCs are under simultaneous pressure from advanced technology scaling and emerging application demands. Technology scaling reduces physical operating margins and makes silicon behavior more variable, less predictable, and more sensitive to physical disturbances. At the same time, AI cloud systems and emerging Physical AI platforms such as autonomous vehicles, robotics, humanoids, drones, and industrial machines demand higher compute density, tighter energy efficiency, and stronger guarantees for functional safety, security, reliability, and in-field integrity. Together, these forces expose the limitations of conventional runtime architectures and make runtime management a first-order architectural challenge rather than a collection of independent subsystem optimizations.

3.1 Technology Scaling Pressure

As semiconductor process technologies continue to scale, the physical behavior of silicon becomes increasingly difficult to predict and manage. Shrinking voltage margins, increasing process variability across dies and local chip regions, higher leakage and static power, stronger sensitivity to power integrity disturbances, localized thermal density, progressive aging and wearout mechanisms such as BTI, HCI, and electromigration, and the growing prevalence of subtle or delay-sensitive manufacturing defects collectively reduce operating margin and increase susceptibility to condition-dependent failures. These effects directly impact runtime timing stability, power efficiency, thermal behavior, and long-term system integrity, making predictable runtime operation significantly harder to sustain across manufacturing variation, workload dynamics, and product lifetime.

3.2 Emerging Application and System Requirements

Modern applications are pushing SoCs into increasingly demanding and less predictable operating regimes. AI cloud infrastructure, large-scale training systems, inference accelerators, and emerging Physical AI platforms such as autonomous vehicles, robotics, drones, and industrial machines require higher compute density, faster data movement, tighter energy

efficiency, and stronger runtime correctness than earlier generations of digital systems. At the same time, many of these platforms must satisfy stringent requirements for functional safety, security, reliability, and in-field diagnostic coverage, particularly in mission-critical deployments where runtime faults, integrity compromise, or gradual degradation can have direct system-level consequences. As a result, modern SoCs are no longer required only to deliver compute performance, but to do so while continuously preserving efficiency, correctness, and system integrity under real operating conditions.

3.3 Legacy and Fragmented Runtime Management in Conventional SoCs

Despite the combined pressures created by technology scaling and emerging system requirements, conventional SoC runtime management remains both fragmented and dependent on legacy techniques. Power management, reliability and aging monitoring, security protection, functional safety, and in-field test have largely evolved as separate engineering domains, each with its own sensing infrastructure, telemetry paths, monitoring logic, and control policies. Rather than operating as elements of a common runtime architecture, these mechanisms are typically deployed as isolated solutions addressing local objectives with limited coordination across the system.

This fragmentation is further compounded by the continued use of conventional methods that are increasingly misaligned with modern silicon behavior. In power management, aging tracking, and certain classes of security monitoring such as thermal or temperature manipulation and aging-oriented attacks, observability often still depends on indirect techniques such as ring oscillators, replica paths, and temperature sensors. These approaches provide only partial and indirect visibility into actual functional silicon conditions and are fundamentally limited in their ability to capture true runtime execution behavior. Functional safety similarly continues to rely heavily on hardware duplication strategies such as lockstep architecture, which impose significant area and power overhead while addressing correctness primarily through redundancy rather than deeper runtime observability. In-field test and continuous structural monitoring also remain largely dependent on heavyweight LBIST and conventional scan-based infrastructures originally developed for manufacturing test rather than efficient, execution-aware runtime use.

As a result, runtime phenomena that are physically interconnected are still monitored and managed through separate mechanisms built on legacy, indirect, or heavyweight techniques that observe different signals, operate on different time scales, and respond according to domain-specific priorities rather than coordinated system-level intent. This increases area, power, routing, and integration overhead, limits system-level visibility, and reduces the ability of conventional SoCs to respond efficiently, cohesively, and scalably to modern runtime challenges.

Table 1 – Conventional Runtime Domains, Common Techniques, and Structural Limitations

Runtime Domain	Common Conventional Techniques	Key Limitations
Power Management	Voltage sensors, temperature sensors, ring oscillators, replica paths, Razor-style / functional path monitoring[1]	Indirect observation of functional timing behavior, limited visibility into localized fast transients, and insufficient alignment with true workload-dependent critical paths
Functional Safety	Lockstep cores[2], error correction codes, watchdogs, safety monitors (Voltage/Clock/Temp. Sensors)	High hardware and power overhead, strong dependence on redundancy, and significant duplication cost for discrepancy detection
Security	Voltage monitors, clock/glitch monitors, temperature monitors, aging/integrity monitors, activity monitors, access-control monitoring, anomaly detection logic	Typically deployed as separate protection structures with fragmented observability, limited cross-domain context, and weak integration with broader runtime control and integrity management
Reliability / Aging	Aging sensors, ring oscillators, canary paths, thermal sensors, margin tracking techniques	Indirect estimation of degradation, limited tracking of true path-level aging behavior, and weak linkage between monitoring results and coordinated runtime mitigation
Diagnostics / In-Field Test	Periodic self-test, BIST, scan-based diagnostics, error logging, health monitors	Intermittent or isolated visibility, disruption or overhead during test operation, and limited integration with continuous runtime management
System-Level Control	Independent domain-specific controllers and monitoring loops	Fragmented observability, duplicated monitoring infrastructure, and inefficient response to tightly coupled runtime events across power, safety, security, reliability, and diagnostics

Notes:

[1] Razor-style techniques remain limited by recovery and hardware overhead, selective deployment scope, data-dependent path activation, incomplete critical-path coverage, clocking complexity, sensitivity to PVT variation, among other practical integration and verification challenges.

[2] Split-lock operation partially mitigates conventional lockstep overhead by allowing independent core operation when full redundancy is not required. However, it retains the underlying duplication cost in protected modes, adds implementation and mode-transition complexity, and reduces the lockstep-based safety coverage available during split operation, requiring additional mechanisms and safety justification for higher ASIL targets.

3.4 Need for a Unified Runtime Paradigm

The limitations of conventional runtime management, combined with the growing pressures of advanced silicon technologies and modern application demands, highlight the need for a more integrated runtime paradigm. Future SoCs require an approach built on new runtime methods and techniques capable of aligning historically disjoint domains—including power management and integrity, functional safety, security, reliability, and diagnostic functions—within a common operational framework. Such a framework must move beyond fragmented sensing and proxy-based estimation toward runtime awareness more directly tied to the functional behavior of silicon itself. It must also support coordinated decision-making across domains, rather than independent subsystem responses to tightly coupled runtime events.

The next section introduces the USRA as this paradigm shift.

4 Unified Silicon Runtime Architecture

Key Observations:

- The architecture is built on consolidated telemetry rather than fragmented sensing infrastructures.
- Runtime phenomena are organized into fast transient and slow varying classes aligned with their temporal behavior.
- In-situ observability within the execution fabric enables direct, functionally relevant runtime visibility.
- A unified runtime framework coordinates observability, evaluation, decision-making, and response across domains.
- The architecture supports both adaptive actuation and integrity or diagnostic outcomes within a common framework.
- The approach is compatible with existing SoC development practices while enabling a new runtime paradigm.

The USRA introduces a new model for runtime system management built around consolidated telemetry, direct observability, and coordinated cross-domain control. Rather than organizing runtime functions as independent subsystems, the architecture establishes a shared framework that captures system behavior within the functional silicon fabric and applies that information across multiple runtime domains.

This unified approach is motivated by the observation that many runtime phenomena affecting modern SoCs are physically interconnected and often manifest through common execution structures. By organizing runtime management around shared observability and coordinated decision-making, the architecture enables a more cohesive and scalable framework for managing power, functional safety, security, reliability, and in-field diagnostics.

4.1 Unified Silicon Runtime Architecture Foundation

The USRA addresses runtime management through a foundational principle: a consolidated telemetry infrastructure that unifies runtime observability across multiple domains. Instead of deploying independent monitoring mechanisms for power management, functional safety, security, and reliability, runtime observability is organized into two fundamental classes: fast transient phenomena and slow-varying phenomena. Fast transient phenomena, such as supply droops and clock glitches, are monitored and mitigated through dedicated detection and correction mechanisms operating in a fast-response loop. In contrast, slow-varying phenomena,

including temperature, aging, and workload-dependent effects, are observed through in-situ instrumentation embedded within the functional execution fabric.

This classification enables a minimal, shared set of telemetry mechanisms that provides meaningful system-wide observability while aligning sensing approaches with the temporal characteristics of the underlying phenomena. By consolidating sensing into these two complementary classes, the architecture reduces redundant infrastructure while maintaining broad coverage and runtime relevance across multiple operating conditions. Such broad in-situ coverage is increasingly important in advanced technology nodes and modern applications, where heightened process, voltage, and temperature variation, together with dynamic workload behavior, can cause indirect or sparsely distributed monitors to diverge from the true runtime behavior of the silicon.

Table 2 – Cross-Domain Coverage of Consolidated Telemetry Infrastructure

Telemetry Mechanism	Power Management	Functional Safety	Reliability / Aging	Security
In-Situ Execution Observability	✓	✓	✓	✓
Supply Droop Detection & Correction	✓	✓		✓
Clock Glitch Detection & Correction		✓		✓

4.2 Unified Silicon Runtime Architecture Overview

Building on this consolidated telemetry foundation, the USRA establishes a runtime framework that integrates observability, analysis, and actuation across the silicon platform. Rather than treating runtime domains as isolated subsystems, the architecture organizes them within a common telemetry and control fabric capable of observing key operational conditions and coordinating system-level response.

Telemetry collected from these observability points is processed by a unified silicon control layer that acts as the runtime intelligence of the system. This control layer interprets telemetry signals, correlates conditions across different operational domains, and determines appropriate system responses. By operating across multiple domains simultaneously, the framework enables coordinated actions that preserve performance, power efficiency, safety integrity, security resilience, and long-term reliability.

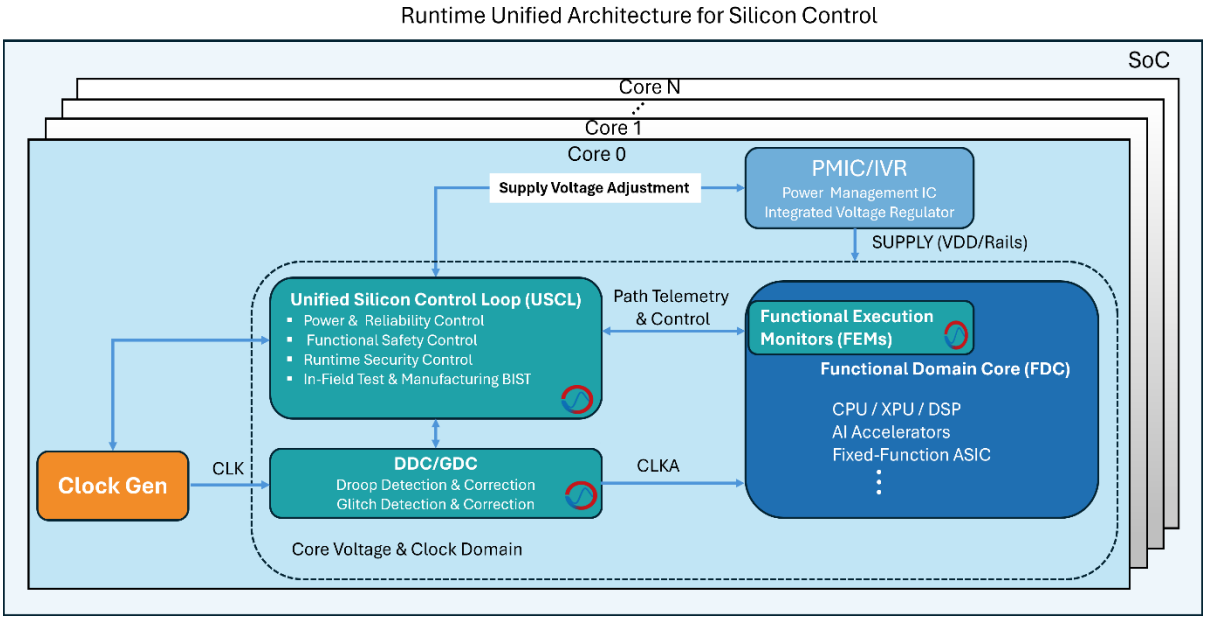


Figure 1 – Unified Silicon Runtime Architecture

To complete the runtime loop, the architecture interfaces with system actuation mechanisms such as PMIC- or regulator-driven voltage control, clock-generation controls, and other protective or adaptive system interfaces. Through these connections, the unified runtime layer can either adjust system operating conditions or produce integrity and diagnostic outcomes based on observed silicon behavior.

4.3 Unified Silicon Runtime Framework

The USRA operates through a runtime framework that enables continuous observation, evaluation, and response across multiple time scales. This framework is decision-driven and supports both adaptive control and integrity-oriented monitoring within a unified model.

At the center of this framework is an in-situ runtime loop that provides deep, context-aware visibility into execution behavior. This loop begins with in-situ observability within the functional execution fabric, followed by runtime activation and execution feedback. The resulting information is processed through runtime evaluation and runtime decision stages. Depending on the observed conditions, the system may either initiate adaptive actuation or produce integrity and diagnostic outcomes such as pass/fail indications, fault status, or runtime anomaly identification.

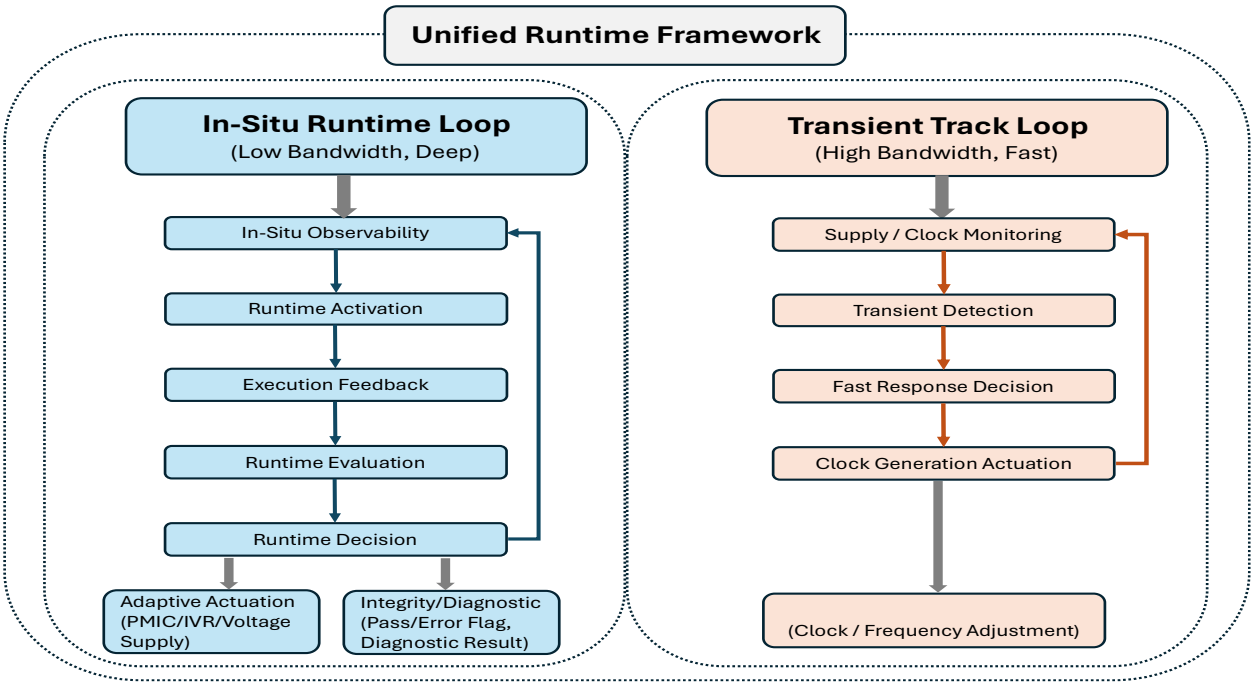


Figure 2 – Unified Runtime Framework

In parallel, the architecture incorporates a fast transient response loop dedicated to supply and clock disturbances. This loop continuously monitors voltage and clock integrity conditions, detects transient anomalies, performs fast response decisions, and applies low-latency corrective action through clock- or frequency-related actuation. This enables rapid stabilization under conditions such as droops, glitches, or other short-duration disturbances.

Together, these two loops form a complementary hierarchical runtime framework. The in-situ loop provides high-fidelity observability and supports deeper runtime evaluation, adaptation, and diagnostics, while the transient loop enables immediate detection and correction of fast integrity events. Both loops continuously feed decision outcomes back into subsequent observation cycles, enabling a resilient and continuously adaptive runtime model.

4.4 Comparison with Conventional Runtime Architectures

The USRA represents a fundamentally different runtime model from conventional approaches. Instead of organizing runtime management around independent domain-specific sensing and control loops, it establishes a shared observability and control framework capable of operating across multiple runtime domains through consolidated telemetry and coordinated decision-making.

Table 3 – Architectural Comparison of Conventional and Unified Runtime Approaches

Aspect	Conventional Runtime Architectures	Unified Silicon Runtime Architecture
Monitoring infrastructure	Independent domain-specific sensors	Consolidated telemetry infrastructure
Observability model	Indirect proxy sensing	In-situ execution observability
Control model	Independent subsystem loops	Unified runtime control
Domain interaction	Limited cross-domain visibility	Coordinated cross-domain management
Scalability	Infrastructure grows per domain	Shared telemetry across domains
Architectural basis	Domain-specific evolution	Cross-domain architectural framework

These differences reflect a shift from fragmented subsystem optimization toward a unified, system-level runtime architecture.

4.5 Unified Runtime Benefits

The USRA translates its foundational principles into measurable benefits across multiple operational domains. By consolidating telemetry, enabling direct observability, and coordinating runtime decisions across domains, the architecture improves both runtime effectiveness and overall system efficiency.

Table 4 – Unified Runtime Architecture: Cross-Domain Benefits

Domain	Conventional Approach	Limitations	Unified Runtime Advantage
Power Management	Guardband-based DVFS, RO/Replica Paths/PMU-based sensing	Over-conservative margins, indirect observability	In-situ observability enables fine-grain adaptive voltage control with reduced guardbands
Safety	Redundant cores, lockstep execution, periodic diagnostics	High area/power overhead, coarse fault coverage	Continuous in-situ validation enables real-time integrity monitoring without full redundancy
Security	Distributed monitors, anomaly detection, software-driven checks	Latency, fragmented visibility, reactive response	Unified observability enables fast detection of abnormal behavior and coordinated response
Reliability (Aging)	Offline characterization, Model-based estimation sensor-based estimation	Poor correlation with real workload, static margins	Runtime tracking of aging through functional observability enables adaptive lifetime management
Test / Diagnostics	Scan-based test, periodic BIST, external testers	High test time, limited in-field coverage	In-field runtime activation and feedback enable continuous structural validation
System Integration	Independent subsystems across power, safety, security, reliability, and diagnostics	Redundant hardware, high complexity, poor coordination	Shared telemetry and a unified decision framework reduce duplication and improve system efficiency
Design Flow Compatibility	Domain-specific solutions often integrated independently	Added integration burden across multiple subsystems	Compatible with conventional SoC design flows and established semiconductor development methodologies without disrupting existing design practices

Taken together, these benefits illustrate that the USRA is not merely a tighter integration of existing runtime functions. It is a new runtime paradigm organized around consolidated

telemetry, direct observability, and coordinated control, enabling modern SoCs to operate more efficiently, accurately, and resiliently under real operating conditions.

5 Runtime Technology Domains Enabled by the Architecture

Key Observations:

- The architecture enables multiple runtime technologies through a common runtime foundation rather than isolated domain-specific mechanisms.
- In-situ observability enables runtime power optimization based on actual silicon behavior rather than conservative guardbands.
- The same observability supports runtime tracking of aging, electromigration, and long-term degradation under real operating conditions.
- Functional integrity and in-field diagnostics can be extended into deployment through direct evaluation within functional logic paths.
- The framework supports both SPFM and LFM objectives while enabling a single-core lockstep-equivalent integrity-monitoring approach.
- Fast transient protection and deeper in-situ monitoring together support coordinated response to runtime fault and attack conditions.
- Runtime security can address both short-duration disturbance attacks and slower physical-layer manipulation strategies.

The USRA enables a broad set of runtime capabilities that have traditionally evolved as separate technologies. Rather than treating power optimization, safety monitoring, security protection, reliability management, and in-field diagnostics as isolated domains, the architecture provides a common runtime basis through which these functions can operate with greater coordination and efficiency. The following subsections outline the key runtime technology domains enabled by this architecture.

5.1 Runtime Power Optimization

Modern SoCs operate under highly dynamic workloads and increasingly tight voltage margins. Achieving minimum-energy operation requires a runtime mechanism capable of identifying the true operating margin of the silicon under real conditions rather than relying on conservative guardbands.

The USRA enables power optimization through in-situ observability of critical execution paths within the functional fabric. The technology allows a large number of critical paths to be selected with negligible area and power overhead, and it can accommodate PVT shifts by selecting different path sets as operating conditions and device characteristics evolve over the lifecycle. These paths are activated without disrupting functional operation and are monitored for setup-time proximity to failure. When timing margin approaches the failure boundary, the runtime framework adaptively adjusts supply voltage to maintain correct operation while minimizing excess margin.

This approach inherently tracks the effects of temperature and aging through the behavior of the monitored paths themselves, eliminating the need to infer margin from indirect environmental proxies. As a result, thermal stress and thermal runaway conditions become directly observable through path behavior, reducing dependence on dedicated temperature sensing for runtime protection decisions. Fast supply disturbances are handled through dedicated droop detection and correction, including intra-clock-cycle response for rapid stabilization. Because the framework is self-adaptive, it does not depend on pre-stored characterization data derived from simulation, lab calibration, or production testing.

This runtime power optimization approach enables:

- Operation at or near true runtime Vmin with reduced conservative guardbands
- Adaptation to PVT shifts across the device lifecycle through dynamic critical-path selection
- Reduced dependence on pre-stored characterization data from simulation, lab calibration, or production testing
- Inherent tracking of temperature and aging through in-situ path behavior, with reduced reliance on dedicated temperature sensing for runtime protection
- Rapid stabilization under supply droops through intra-clock-cycle correction, improving power efficiency while preserving stability and functional correctness

5.2 Runtime Reliability and Aging Management

Modern silicon platforms must maintain reliable operation over extended lifetimes while operating under sustained electrical, thermal, and workload-induced stress. As advanced technology nodes reduce timing margins and increase sensitivity to degradation mechanisms, reliability management can no longer depend solely on static design assumptions or sparse proxy monitors.

The USRA enables runtime reliability and aging management through in-situ observability within the functional execution fabric. Critical execution paths naturally reflect gradual shifts in silicon behavior caused by aging, electromigration, thermal stress, and long-term operating conditions. By observing these paths during normal operation, the runtime framework gains direct visibility into evolving timing margins across the active silicon domain.

Because the monitored structures are part of the functional fabric, they provide a more representative view of real degradation than indirect sensors or offline estimates. This allows the architecture to detect progressive margin shifts associated with long-term degradation and workload-dependent wear over the device lifecycle. The framework can also adapt to lifecycle evolution through dynamic selection of different critical path sets as degradation patterns and operating conditions change over time.

This runtime reliability and aging management approach enables:

- Continuous visibility into timing-margin shifts caused by aging, electromigration, thermal stress, and long-term operating conditions
- Direct observation of degradation within the functional execution fabric rather than through indirect proxy estimates
- Broader and more relevant coverage than a limited set of discrete aging or thermal sensors
- Adaptation to lifecycle-driven changes through dynamic selection of different critical path sets
- Early identification of localized degradation trends before they affect functional correctness
- Maintenance of safe operating margins while preserving performance and long-term stability
- Long-term reliability management without dependence on static lifetime assumptions or sparse offline characterization

5.3 Runtime Functional Integrity, Safety, and In-Field Diagnostics

Modern silicon platforms deployed in safety-sensitive and mission-critical environments must preserve functional correctness during operation while also providing diagnostic coverage for both active and latent faults. Conventional approaches often address these requirements through heavyweight redundancy and periodic diagnostic testing, but modern SoCs increasingly require runtime mechanisms that support integrity monitoring and in-field validation within a more scalable and proactive framework.

The USRA enables runtime functional integrity, safety, and in-field diagnostics through direct observability and controllability of functional logic under real operating conditions. This allows the architecture to support both single-point fault metric (SPFM) and latent fault metric (LFM) objectives within a unified runtime model, aligning naturally with the needs of safety-oriented and mission-critical systems. Rather than relying solely on separate diagnostic infrastructures or full compute duplication, the architecture enables a single-core lockstep-equivalent approach in which fault effects can be observed directly within functional logic paths during runtime operation.

By applying a novel runtime approach to exercising functional logic paths without disrupting normal operation, the framework can detect deviations in expected behavior, identify runtime anomalies, and extend diagnostic coverage into deployment. This approach is proactive, enabling fault conditions relevant to both SPFM and LFM objectives to be detected before they propagate into visible functional failure. The same runtime activation and execution feedback mechanisms can also reveal latent or emerging faults that may not be exposed through normal operation alone, creating a unified path for continuous integrity monitoring and in-field diagnostic assessment.

In addition, transient integrity-threatening conditions such as supply droops are addressed through dedicated droop detection and correction mechanisms operating in the fast response loop. This allows such events to be detected and mitigated before they propagate into functional

failure, complementing the deeper in-situ monitoring used for SPFM- and LFM-relevant coverage. As a result, the architecture combines long-horizon diagnostic coverage with rapid transient protection in a form well aligned with the expectations of mission-critical runtime monitoring.

Beyond enabling a single-core lockstep-equivalent approach, the technology can also complement conventional dual-core lockstep systems by helping mitigate inherent common-cause failure (CCF) limitations. Through broader runtime observability and complementary diagnostic coverage, the architecture can improve visibility into fault conditions that may simultaneously affect redundant execution paths.

This runtime functional integrity, safety, and in-field diagnostic approach enables:

- A single-core lockstep-equivalent approach to runtime integrity monitoring with reduced architectural overhead
- Support for both SPFM and LFM objectives within a common runtime framework
- Extension of diagnostic coverage into deployment without dependence on separate test-only infrastructures
- Proactive detection of fault conditions before visible functional failure
- Mitigation of transient integrity threats such as supply droops through fast detection and correction
- Complementary enhancement of conventional dual-core lockstep through improved visibility into common-cause failure conditions

5.4 Runtime Security Monitoring

Modern silicon platforms are increasingly exposed to malicious techniques that attempt to disrupt correct system behavior by targeting the physical operating conditions of the device. These attacks may seek to induce faults, alter execution behavior, or compromise system integrity through voltage, clock, thermal, or aging-related manipulation. As cybersecurity becomes a pivotal requirement in modern connected, intelligent, and mission-critical systems, runtime security must include mechanisms capable of detecting both fast transient attacks and slower physical-layer manipulation strategies.

The USRA enables runtime security monitoring across both classes of threat. Transient attacks such as power-supply manipulation through undervolting and malicious clock attacks are addressed through dedicated droop detection and correction and clock-glitch detection and correction mechanisms operating in the fast response loop. This enables low-latency detection and mitigation of short-duration disturbance attacks before they propagate into functional corruption.

Slower physical-layer manipulation strategies such as temperature- and aging-related attacks are addressed through in-situ observability within the functional execution fabric. Because these monitored paths directly reflect execution-relevant silicon behavior, the architecture can detect

abnormal conditions and fault-inducing effects that would not be accurately represented by sparse proxy sensors or averaged environmental measurements.

This runtime security monitoring approach enables:

- Low-latency detection and mitigation of undervolting and malicious clock attacks
- Detection of temperature- and aging-related manipulation through in-situ functional path observability
- Coordinated protection against both fast transient and slow physical-layer attack strategies
- Broader and more functionally relevant visibility than sparse proxy-based security monitors
- Reduced dependence on isolated security-specific sensing infrastructures
- Improved runtime integrity protection against both immediate and long-horizon attack conditions

6 System-Level Impact of the Unified Silicon Runtime Architecture

Key Observations:

- Improved power efficiency through reduced guardbands and lower runtime monitoring overhead
- Improved silicon area utilization through consolidation of monitoring and support infrastructure
- Reduced reliance on duplicated compute and protection resources in safety-oriented architectures
- Simplified system integration through reduced subsystem fragmentation and lower development complexity

The USRA improves system efficiency and resilience through both architectural innovation and the consolidation of runtime observability and control across traditionally independent operational domains. By replacing fragmented monitoring and control subsystems with a shared runtime framework, the architecture improves power efficiency, silicon area utilization, system performance, and overall design efficiency through enhanced runtime management and reduced infrastructure duplication.

6.1 Power Efficiency Improvements

The USRA improves power efficiency through several complementary mechanisms spanning runtime power control, monitoring infrastructure consolidation, safety architecture optimization, and runtime diagnostic techniques.

Table 5 – Power Efficiency Improvements Enabled by the USRA

Power Reduction Mechanism	Architectural Technique	Impact
Operation near true critical Vmin	• In-situ timing observability • Droop detection and correction	• Enables operation closer to the true critical voltage (Vmin) • Reduces guardbands traditionally required for droop events, aging effects, and worst-case process variability
Monitoring infrastructure consolidation	• Shared telemetry infrastructure • Unified runtime control across multiple domains	• Reduces the number of distributed monitoring elements • Lowers power consumed by monitoring infrastructure
Single-core lockstep-equivalent safety	• Architectural techniques enabling lockstep-equivalent protection on a single core	• Reduces or eliminates duplicated compute resources required by conventional dual-core lockstep architectures
Reduced reliance on scan-based and LBIST testing	• Alignment of manufacturing and in-field diagnostic methodologies • Continuous runtime observability of functional logic	• Reduces switching activity associated with periodic test operations • Lowers leakage and dynamic power overhead associated with test infrastructure • Relax some timing and margin constraints otherwise imposed by test-oriented structures

6.2 Silicon Area Reduction

Silicon area overhead associated with runtime monitoring and protection mechanisms has increased significantly in modern SoCs. Conventional architectures often rely on numerous dedicated circuits for sensing, safety monitoring, power integrity tracking, and structural diagnostics, resulting in duplicated hardware across multiple runtime subsystems.

The USRA reduces this overhead through both architectural innovation and consolidation of observability and control mechanisms within the silicon execution fabric. By reusing runtime infrastructure across multiple operational domains and enabling more efficient safety and monitoring architectures, the approach improves silicon resource utilization. Reduced silicon area not only simplifies on-chip infrastructure, but can also improve fabrication yield and lower manufacturing and packaging costs.

Table 6 – Silicon Area Reduction Enabled by the USRA

Area Reduction Mechanism	Architectural Technique	Impact
Monitoring infrastructure consolidation	Shared telemetry infrastructure and unified runtime control across multiple domains	• Reduces the number of distributed monitoring circuits • Simplifies monitoring logic across runtime subsystems
Single-core lockstep-equivalent safety	Architectural techniques enabling lockstep-equivalent protection on a single core	• Reduces or eliminates duplicated compute cores required by conventional dual-core lockstep architectures
Reduced reliance on scan and LBIST infrastructure	Alignment of manufacturing and in-field diagnostics through continuous runtime observability	• Reduces the extent of scan logic, scan clocks, and test control infrastructure • Reduce reliance on LBIST during In-Field Testing
Droop detection and correction	Fast transient detection combined with runtime correction mechanisms (Fast Droop mitigated locally)	• Reduces reliance on large decoupling capacitor networks • Enables less aggressive PDN design • Reduce voltage regulation overhead (lower phase count / Low bandwidth)
In-situ observability within functional logic	Monitoring embedded within the functional execution fabric	• Avoids deployment of large networks of dedicated monitoring circuits (Often analog heavy)
Overall infrastructure consolidation	Consolidation of runtime monitoring and protection mechanisms across domains	• Improves silicon utilization efficiency • Improve fabrication yield • Lower manufacturing and packaging cost

6.3 System Integration Efficiency

Beyond improvements in power efficiency and silicon area utilization, the USRA simplifies overall system design by reducing fragmentation across runtime monitoring and control functions. These architectural simplifications can reduce complexity across design, verification, integration, test, firmware, and software development while improving overall development efficiency.

7 Implications for Future Silicon Platforms

The continued evolution of AI-driven and heterogeneous computing systems is increasing the demand for silicon platforms that can operate efficiently, securely, and reliably under highly dynamic conditions. At the same time, advanced process scaling and shrinking operating margins are making static management approaches increasingly inadequate.

In this environment, future silicon platforms will require runtime architectures capable of continuously observing system behavior and coordinating responses across multiple operational domains. The USRA points toward this direction by providing a scalable foundation for adaptive runtime management across power, safety, security, and reliability.

As SoCs continue to integrate CPUs, AI accelerators, and specialized processing elements, unified runtime control will become increasingly important for maintaining efficiency, resilience, and system-level coordination across complex silicon platforms.

8 Conclusion

Modern SoCs are being pushed simultaneously by advanced technology scaling and increasingly demanding AI-era applications. As operating margins shrink and runtime behavior becomes more dynamic, fragmented runtime architectures built around isolated sensing and control mechanisms become progressively less accurate, less efficient, and less scalable. At the same time, many existing runtime solutions remain constrained by legacy techniques that are only incrementally adapted to modern challenges, despite increasingly tighter coupling between power, timing, thermal behavior, aging, safety, security, and diagnostics. In this context, runtime management can no longer remain a collection of domain-specific subsystems layered onto the silicon platform; it must become a first-class architectural capability.

This paper introduced the Unified USRA as a new runtime paradigm built around consolidated telemetry, in-situ observability within the functional execution fabric, and coordinated cross-domain control. By organizing runtime management through a shared framework rather than independent subsystem-specific mechanisms, the architecture enables power optimization, reliability and aging management, functional integrity and safety monitoring, in-field diagnostics, and runtime security through a common silicon-aware control model.

The significance of this approach extends beyond improved monitoring or tighter integration. It establishes a new architectural model for runtime management in which observability is tied directly to functionally relevant behavior, decisions are coordinated across domains, and system response is aligned with both fast transient events and longer-horizon operating conditions. This enables measurable improvements in power efficiency, performance stability, silicon area utilization, safety and security resilience, and overall system integration efficiency.

As computing platforms continue to evolve toward increasingly heterogeneous, AI-driven, and mission-critical systems, runtime intelligence will become a foundational requirement of silicon design. The USRA represents a step toward that future by replacing fragmented runtime management and limited legacy adaptation with a unified silicon-aware control framework capable of continuously observing, adapting, and protecting system behavior under real operating conditions.