Unified Silicon Runtime™ Test Architecture

From Manufacturing Test to Runtime Structural Diagnostics

Technical Foundations and Deployment Path for VegaTest™

Factory Test • In-Field Diagnostics • SLM • RAS

White Paper | May 2026

Ben Zaryouh
Founder & CEO
VegaSemi
ben@vegasemi.com

Table of Contents

1 Executive Summary 1

2 Introduction 2

3 Limitations of Conventional SoC Test 3

3.1 Factory Test — Structural but Static 3

3.2 System-Level Test (SLT) — Realistic but Not Fault-Model-Driven 4

3.3 In-Field Techniques — Observable but Not Fault-Grade 6

3.4 Monitoring and Telemetry-Based Approaches 8

3.5 The Missing Runtime Structural Validation Framework 9

4 VegaTest™ Architectural Paradigm 12

4.1 Unified Runtime Structural Diagnostics and Orchestration 13

4.2 Architectural Impact and Deployment Benefits 15

5 Why This Matters Now 17

6 Conclusion 19

1 Executive Summary

Modern SoC test is becoming increasingly fragmented, expensive, and disconnected from real runtime behavior. Factory test provides structural coverage, but only under controlled manufacturing conditions. SLT adds realism, but lacks formal fault models and measurable structural completeness. In-field diagnostics extend testing into deployment, but often remain scan-centric, test-mode dependent, or limited in runtime fidelity. Monitoring improves visibility, but relies on indirect, correlation-dependent signals rather than deterministic structural validation.

These limitations are becoming more critical as advanced nodes, AI workloads, automotive systems, chiplets, aging effects, and silent data corruption create failure mechanisms that depend on real workload, thermal, voltage, timing, and lifecycle conditions. At the same time, conventional test flows face growing pressure from large ATPG vector volumes, long ATE time, scan-heavy infrastructure, package/GPIO overhead, and expensive SLT and monitoring environments.

VegaTest™ introduces a unified runtime structural diagnostics architecture that aligns factory test, system-level diagnostics, in-field validation, SLM, RAS, and reliability within a continuous silicon integrity framework. The architecture combines structural diagnostic rigor, runtime-aware orchestration, execution-coupled validation, and coordinated factory-to-field deployment while reducing reliance on scan-heavy infrastructure and fragmented test flows.

VegaTest is not a replacement for conventional DFT methodologies; it extends them into a runtime-aware structural diagnostics framework. This architectural shift improves test quality, reduces escapes and silent data corruption risk, lowers vector and ATE burden, improves yield and binning, and strengthens long-term silicon integrity across production, deployment, and operational lifetime.

2 Introduction

Semiconductor validation is evolving beyond traditional manufacturing-centric methodologies. Advanced-node SoCs increasingly operate under dynamic workload, thermal, voltage, timing, and lifecycle conditions that are difficult to fully capture through isolated factory test environments alone. At the same time, AI infrastructure, heterogeneous compute platforms, automotive systems, and mission-critical deployments are increasing the importance of runtime correctness, long-term reliability, and silent data corruption resilience across the operational lifetime of silicon.

These trends are driving a broader industry transition toward System-Level Test, in-field diagnostics, runtime observability, Silicon Lifecycle Management, and continuous reliability monitoring. However, existing methodologies remain fragmented between structural validation, runtime operation, and lifecycle awareness, while the associated cost and infrastructure burden of conventional test flows continue to grow.

This paper examines the growing gap between manufacturing-time structural validation and real runtime behavior. It reviews the limitations of existing factory test, SLT, in-field diagnostics, and monitoring approaches, then introduces VegaTest™ as a unified runtime structural diagnostics architecture spanning manufacturing, deployment, and operational lifetime.

3 Limitations of Conventional SoC Test

Key Observations

- Structural validation is primarily achieved during manufacturing, but does not extend into real system operation.
- System-Level Test (SLT) provides realistic execution conditions, yet lacks deterministic fault models and measurable structural coverage.
- In-field techniques enable runtime observability, but rely on indirect indicators and do not guarantee structural correctness.
- Monitoring-based approaches increase visibility, but remain correlation-dependent and structurally incomplete.
- No existing methodology provides **structural validation during real operation with deterministic and measurable coverage across the silicon lifecycle**.

3.1 Factory Test — Structural but Static

Production test remains the primary mechanism for achieving structural validation in modern SoC design. Scan-based methodologies, combined with at-speed techniques such as Transition Delay Fault and Path Delay Fault, provide high fault coverage under well-defined fault models. These approaches enable deterministic validation and quantifiable coverage at the point of manufacturing.

However, this validation is inherently static.

Test conditions are applied in controlled environments that do not reflect real system behavior. While structural faults are targeted through modeled conditions, the activation of timing-critical paths is limited to pre-generated patterns rather than actual execution scenarios. As a result, many context-dependent or workload-sensitive conditions remain unexercised.

This limitation becomes more pronounced as SoCs increase in complexity. Achieving high coverage requires large test vector volumes, increasing test time and cost. At the same time, a fundamental tradeoff persists between yield and defect escape: aggressive screening risks yield loss, while relaxed testing increases the likelihood of latent defects reaching the field.

Most importantly, structural validation effectively stops after manufacturing. Once deployed, changes in workload, thermal behavior, voltage conditions, and aging are no longer covered by the original structural test framework.

> **Factory test provides strong structural validation at manufacturing time, but remains costly, vector-intensive, ATE-dependent, and disconnected from real runtime conditions. This limits visibility into field behavior, margin evolution, yield/binning optimization, and long-term silicon integrity.**

3.2 System-Level Test (SLT) — Realistic but Not Fault-Model-Driven

System-Level Test (SLT) has become an essential component of modern semiconductor validation flows. As SoC complexity increases and application behavior becomes more dynamic, traditional manufacturing test alone is no longer sufficient to expose all relevant failure mechanisms. Industry adoption reflects this shift, with SLT now widely deployed as a post-manufacturing validation stage for high-performance processors, AI accelerators, automotive systems, and other complex devices.

SLT evaluates devices under conditions that closely resemble real-world operation. This includes execution of software workloads, validation of hardware–software interactions, and system-level behavior across power, thermal, clock, and communication domains. Unlike conventional Automated Test Equipment (ATE), which focuses on structural and parametric validation of individual blocks, SLT exercises the complete system in its intended usage context. This enables detection of failure modes that depend on cross-domain interactions, workload-induced stress, transient operating conditions, and complex software execution behavior.

This realism provides a critical advantage. SLT can reveal issues that are difficult or impossible to detect through structural test alone, including marginal timing behavior, interface interoperability problems, thermal-induced instabilities, workload-sensitive failures, and silent data corruption scenarios. As a result, SLT has become increasingly important for advanced-node devices, AI processors, heterogeneous systems, and mission-critical applications where near-zero defect escape targets are required.

However, the same characteristics that make SLT effective also define its fundamental limitation.

SLT is not built around a formal structural fault model. Unlike scan-based ATPG methodologies, which target explicitly defined fault classes and produce measurable coverage metrics, SLT relies on workloads, benchmarks, stress scenarios, and application behavior whose relationship to underlying structural faults is not deterministic. The activation of critical logic paths depends

on software behavior, system configuration, workload sequencing, and environmental conditions, and therefore cannot be systematically guaranteed.

As a result, structural coverage cannot be directly quantified. While SLT can demonstrate that a device behaves correctly under a given set of operating conditions, it does not provide assurance that all relevant structural fault conditions have been exercised. Different workloads activate different portions of the design, leaving portions of the structural space either untested or only partially explored.

In addition, SLT behavior is inherently non-deterministic. Test outcomes may vary depending on workload intensity, execution duration, thermal conditions, system state, and data patterns. Increasing workload stress or runtime duration may expose new issues, while reproducing failures can be difficult because the exact operating context that triggered the failure is not always controllable or observable. This complicates debug, root-cause analysis, and correlation to specific structural conditions.

SLT also introduces practical challenges. It requires complex infrastructure combining hardware platforms, software environments, thermal management systems, automation frameworks, and workload orchestration. Test development is often empirical and workload-driven, leading to long execution times, increased cost, and scalability limitations.

From a validation perspective, SLT operates primarily as a screening mechanism. It can reveal the presence of failures under realistic conditions, but does not inherently identify their structural origin or guarantee completeness of testing. It bridges the gap between manufacturing validation and real-world behavior, but does not close it.

> **SLT reveals failure symptoms under realistic operating conditions, but without a fault model it cannot quantify structural correctness.**

Recent industry work has begun extending structural ATPG techniques beyond traditional ATE environments into system-level and in-field deployments. These approaches recognize an important industry reality: factory-only structural test is no longer sufficient for advanced SoCs, and structural diagnostics increasingly need to move closer to deployed operating environments.

However, many of these solutions remain fundamentally scan-based. In effect, they relocate portions of the ATE function into an SLT or in-field environment by transporting structural patterns through embedded controllers or high-speed interfaces, while continuing to rely on scan shift/capture behavior, controlled test modes, and scan-oriented pattern execution.

As a result, the fundamental limitations of scan-centric execution remain largely unchanged:

- dependence on scan infrastructure and scan-access mechanisms
- controlled test-mode execution rather than true mission-mode operation
- switching activity that differs from real functional workloads
- thermal, voltage, droop, and hotspot profiles that may not reflect runtime conditions
- brief at-speed capture events rather than continuous workload-driven execution

Structural coverage may remain fault-grade in content, but the execution environment itself may still differ substantially from the real operating conditions under which workload-sensitive, intermittent, or marginal failures emerge.

> **These approaches move structural test closer to the deployed system, but remain scan-centric and test-mode dependent, falling short of runtime-faithful silicon diagnostics.**

3.3 In-Field Techniques — Observable but Not Fault-Grade

As semiconductor devices increasingly operate in mission-critical environments, validation is no longer confined to manufacturing and bring-up phases. Automotive systems, AI accelerators, data-center processors, networking infrastructure, and edge-computing platforms are now expected to maintain high levels of correctness and reliability throughout their operational lifetime. This has driven growing industry adoption of in-field structural test techniques capable of detecting latent defects, degradation effects, and operational failures after deployment.
Common in-field test approaches include Logic Built-In Self-Test, Memory Built-In Self-Test, scan-assisted diagnostics, and software-driven test routines. These techniques provide important capabilities for startup diagnostics, periodic health monitoring, maintenance validation, and safety-oriented fault detection during product lifetime.
LBIST in particular has become widely adopted because it enables autonomous structural logic testing without requiring full external ATE access. By integrating on-chip pattern generation and response compaction logic, LBIST allows structural test capability to be deployed directly within fielded systems. MBIST similarly provides strong coverage for embedded memory arrays and remains a foundational component of modern in-field diagnostics.
However, while LBIST is valuable, pseudo-random LBIST alone is increasingly insufficient for critical applications.
Modern SoCs contain highly constrained control logic, timing-sensitive structures, deep sequential behavior, and rare fault conditions that may not be efficiently activated by pseudo-random stimulus. Achieving acceptable coverage often requires additional test points, pattern tuning, longer execution time, and increasingly complex DFT infrastructure. This limitation has become especially important for automotive, AI, networking, and data-center systems where defect escape targets, functional safety requirements, and silent data corruption concerns demand significantly higher diagnostic confidence.
In addition to coverage limitations, LBIST introduces non-trivial overhead and operational constraints, including:

- pattern generation and response compaction logic
- additional DFT control infrastructure
- power and thermal distortion caused by pseudo-random switching activity
- verification and integration complexity
- interruption of normal operation through dedicated diagnostic windows or test modes

Most importantly, LBIST execution conditions often differ substantially from true mission-mode behavior. In many systems, in-field structural testing is scheduled during startup, shutdown, idle windows, maintenance intervals, or controlled diagnostic modes rather than during active workload execution. As a result, the thermal, voltage, workload, hotspot, and power-integrity conditions experienced during test may differ significantly from those encountered during real operation.

This distinction is critical because many modern failure mechanisms are highly runtime-dependent. Marginal timing paths, localized droops, transient effects, workload-sensitive behavior, and intermittent faults may emerge only under specific execution conditions that are difficult to reproduce using pseudo-random or scan-oriented structural activity.

In response to these limitations, the industry has begun moving toward deterministic in-system structural testing. Recent approaches extend ATPG-generated deterministic patterns beyond traditional ATE environments into deployed systems using embedded controllers, scan networks, and high-speed system interfaces. This evolution reflects an important industry recognition:

Factory-only structural testing is no longer sufficient for advanced SoCs.

Deterministic in-system testing improves structural test quality compared to pseudo-random-only LBIST approaches and enables higher-confidence diagnostic screening in deployed environments. However, many of these solutions remain fundamentally scan-centric. In effect, they relocate portions of the ATE function into the deployed system while continuing to rely on scan shift/capture behavior, scan-access infrastructure, controlled test modes, and scan-oriented execution.

As a result, several core limitations remain:

- dependence on scan infrastructure and scan-access mechanisms
- controlled test execution rather than true mission-mode operation
- switching activity that differs from real functional workloads
- thermal, voltage, droop, and hotspot behavior that may not reflect runtime conditions
- brief at-speed capture events rather than continuous workload-driven execution

Structural coverage may therefore remain fault-grade in content while the execution environment itself still differs substantially from the runtime conditions under which intermittent, workload-sensitive, or marginal failures actually emerge.

> **The industry is clearly moving structural diagnostics closer to deployed operation. However, existing in-field approaches still do not fundamentally transform structural testing into runtime-faithful silicon diagnostics.**

3.4 Monitoring and Telemetry-Based Approaches

Recent approaches have focused on enhancing silicon visibility through embedded monitoring infrastructure, distributed telemetry, and data-driven analytics. These techniques instrument the design with sensors and agents that capture parametric indicators such as timing behavior, voltage fluctuations, thermal activity, aging effects, and interconnect performance across different operating conditions.

By collecting data during manufacturing, system-level test, and field operation, these approaches aim to provide a continuous view of silicon behavior throughout its lifecycle. When combined with analytics and machine learning models, this data can be used to infer performance margins, detect anomalies, and predict potential reliability issues.

This represents a meaningful advancement in observability compared to traditional test methodologies. It enables visibility into dynamic behavior that would otherwise remain hidden and supports new use cases in system monitoring, predictive maintenance, and lifecycle management.

However, this approach remains fundamentally indirect.

The signals captured by monitoring infrastructure are not direct representations of structural correctness. Instead, they serve as proxies that must be interpreted through correlation models, calibration data, and statistical inference. The relationship between a given sensor reading and the underlying structural condition of the silicon is not deterministic, and may vary across workloads, operating conditions, and device aging.

As a result, these approaches do not provide fault-model-driven validation. Unlike structural test methodologies, which target explicitly defined fault classes and produce measurable coverage metrics, monitoring-based techniques cannot guarantee that all relevant structural fault conditions have been exercised or observed. Coverage is inherently approximate and dependent on sensor placement, model fidelity, and the completeness of observed data.

Furthermore, achieving high observability often requires significant infrastructure, including distributed sensors, data aggregation networks, and analytics pipelines spanning on-chip, tester, and system environments. This increases design complexity, introduces area and power overhead, and requires ongoing calibration and model maintenance across the product lifecycle.

In addition, correlation fidelity may degrade over time. Variations in workload, localized thermal effects, voltage noise, and aging mechanisms can alter the relationship between monitored signals and actual silicon behavior, making consistent interpretation increasingly challenging.

> **Monitoring-based approaches improve visibility into silicon behavior, but remain indirect and correlation-dependent rather than deterministic structural validation frameworks. Observability alone does not guarantee structural correctness.**

3.5 The Missing Runtime Structural Validation Framework

Across all existing approaches, a consistent pattern emerges:

- Factory test provides structural coverage, but only during manufacturing under controlled conditions.
- System-Level Test provides realistic execution behavior, but without formal fault models or measurable structural completeness.
- In-field techniques extend diagnostics into deployed systems, but remain constrained by scan-oriented execution, controlled test modes, or limited runtime fidelity.
- Monitoring and telemetry approaches improve observability, but rely on indirect, correlation-dependent indicators rather than deterministic structural validation.

Each approach addresses an important dimension of the problem, yet none provides a unified framework capable of maintaining structural integrity awareness continuously across the silicon lifecycle.

This fragmentation creates more than methodological inefficiency; it creates diagnostic discontinuity. Factory test, SLT, in-field diagnostics, and monitoring are often developed, deployed, and interpreted as separate layers, each with different assumptions, observables, coverage models, and execution conditions. As a result, defects or marginal conditions that escape one layer are not necessarily exposed by the next.

A path that is not sensitized during scan may not be exercised by SLT. A workload-dependent failure may not appear during scheduled in-field diagnostics. A localized degradation mechanism may be observed only indirectly through telemetry and monitoring infrastructure. This discontinuity increases the risk of latent defects, intermittent failures, silent data corruption, and costly field escapes.

At the same time, the cost and infrastructure burden associated with existing methodologies continues to increase.

Modern scan-based structural test flows require:

- extensive scan insertion infrastructure
- large ATPG vector volumes
- long test execution times
- expensive high-performance ATE systems
- high-bandwidth test access mechanisms
- additional GPIO and package overhead for test access

- sequential test execution across increasingly complex SoC structures

As devices scale, these costs grow substantially across manufacturing, validation, SLT, and in-field deployment flows.

Similarly, advanced runtime monitoring approaches often require:

- distributed sensors and agents
- telemetry aggregation networks
- analytics infrastructure
- calibration and model maintenance
- additional area and power overhead

The industry itself is increasingly moving in this direction. The growing adoption of SLT, in-field diagnostics, deterministic in-system structural testing, and lifecycle monitoring reflects a broad recognition that factory-only validation is no longer sufficient for modern SoCs.

However, current approaches remain fragmented between:

- structural but non-runtime-aware methodologies
- runtime-aware but non-structural methodologies
- observable but non-deterministic methodologies

As a result, a fundamental gap remains unresolved.

What is missing is a capability that combines:

- structural validation
- runtime operation
- deterministic, fault-grade coverage
- execution conditions aligned more closely with real functional behavior
- reduced dependence on heavy scan, tester, and monitoring infrastructures

within a single continuous framework spanning manufacturing, deployment, and field operation.

Modern failure mechanisms increasingly emerge under highly dynamic conditions driven by:

- workload-dependent activity
- localized thermal behavior
- transient power integrity effects
- execution-dependent timing behavior
- aging and lifecycle degradation

These conditions are difficult to reproduce using conventional structural test modes, pseudo-random diagnostics, or indirect monitoring alone.

Consequently, structural correctness and runtime behavior remain only partially connected in existing validation methodologies, while the associated cost, infrastructure complexity, and deployment overhead continue to grow.

> **No existing approach provides deterministic structural diagnostics continuously aligned with real runtime operation across the silicon lifecycle while simultaneously reducing the growing cost, fragmentation, and infrastructure burden of conventional test and monitoring methodologies.**

Addressing this gap requires a fundamentally different architectural direction: extending structural diagnostics beyond isolated manufacturing test events and toward runtime-aware, execution-coupled silicon integrity mechanisms.

4 VegaTest™ Architectural Paradigm

Key Observations

- VegaTest™ introduces a unified runtime structural diagnostics architecture spanning factory test, system-level diagnostics, in-field validation, SLM, RAS, and reliability.
- The architecture fuses structural diagnostic intent with runtime-aware execution, moving test closer to real workload, thermal, voltage, timing, and aging conditions.
- VegaTest targets major cost drivers in conventional test flows, including scan-heavy infrastructure, large ATPG vector volumes, long ATE time, package/GPIO overhead, and fragmented validation stages.
- By aligning factory and field diagnostics within one lifecycle framework, VegaTest improves test quality, reduces escapes, supports better yield/binning, and strengthens long-term silicon integrity.

Conventional semiconductor validation methodologies remain fundamentally fragmented. Manufacturing test, System-Level Test, in-field diagnostics, telemetry, and lifecycle monitoring each address important aspects of silicon validation, yet they operate through different infrastructures, assumptions, execution conditions, and observability models.

As a result, structural integrity awareness becomes discontinuous across the silicon lifecycle. Structural validation remains concentrated at manufacturing time, while runtime behavior, workload-dependent conditions, thermal effects, transient power integrity behavior, intermittent timing conditions, aging-related degradation, and silent data corruption mechanisms remain only partially connected to deterministic structural diagnostics.

At the same time, the cost and infrastructure burden of conventional methodologies continues to increase through:

- extensive scan insertion
- large ATPG vector volumes
- long ATE execution time
- package and GPIO overhead
- distributed monitoring infrastructure
- fragmented validation and deployment flows

VegaTest™ introduces a different architectural direction.

Rather than treating factory test, SLT, in-field diagnostics, SLM, RAS, and reliability monitoring as isolated layers, VegaTest establishes a unified runtime structural diagnostics framework spanning manufacturing, deployment, and operational lifetime.

The architecture combines:

- structural diagnostic rigor
- runtime-aware orchestration

- execution-coupled validation
- embedded diagnostic control
- coordinated factory-to-field deployment

within a continuous silicon integrity framework.

At the architectural level, VegaTest can be viewed as a fusion of structural and functional validation principles. Rather than treating structural coverage and runtime execution as disconnected domains, VegaTest aligns them within a coordinated framework capable of maintaining stronger structural integrity awareness throughout real system operation.

4.1 Unified Runtime Structural Diagnostics and Orchestration

VegaTest extends structural diagnostics beyond isolated manufacturing events and toward a unified silicon lifecycle framework. Rather than treating factory test, system-level diagnostics, in-field validation, SLM, RAS, and reliability as separate layers, VegaTest aligns them within a coordinated runtime structural diagnostics architecture.

As illustrated in Figure 1, VegaTest integrates manufacturing test, system-level diagnostics, in-field validation, runtime structural integrity assessment, and lifecycle monitoring within a continuous deployment model.

Figure 1 — VegaTest™ Unified Runtime Structural Diagnostics Architecture

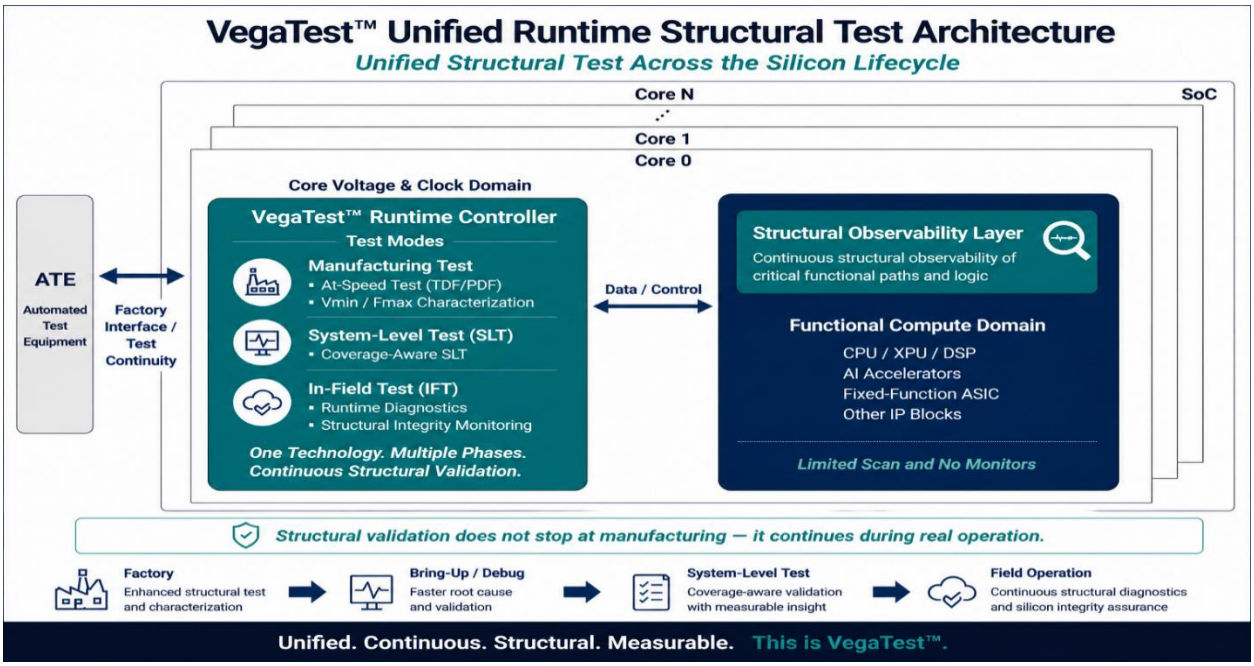

Figure 1. *VegaTest™ architectural framework integrating manufacturing test, system-level diagnostics, in-field structural integrity, and execution-coupled runtime observability within a unified silicon lifecycle deployment model. The architecture combines adaptive runtime control, functional-domain observability, and coordinated factory-to-field diagnostics while reducing dependence on conventional scan-centric infrastructures.*

Key Architectural Characteristics

- Unified factory-to-field structural diagnostics
- Runtime-aware structural integrity assessment
- Reduced reliance on pervasive scan-heavy infrastructure
- Coordinated SLM, RAS, and reliability diagnostics
- Functional-domain aligned observability and control

Within this framework, structural integrity awareness is no longer confined to manufacturing-time validation alone. Instead, structural diagnostics become more continuous, deployment-aware, and aligned with the conditions that matter across the silicon lifecycle.

A key architectural principle of VegaTest is runtime-aware diagnostic orchestration. Traditional structural methodologies primarily exercise silicon through scan activity, controlled capture windows, pseudo-random switching behavior, or isolated diagnostic modes. These approaches can provide strong manufacturing-time structural coverage, but their execution environment may differ substantially from real workload operation.

VegaTest introduces a proprietary runtime diagnostic architecture that strengthens structural integrity assessment while reducing reliance on scan-heavy infrastructure and fragmented test-mode execution.

At the architectural level, VegaTest combines:

- selected structural diagnostic targets
- proprietary runtime orchestration techniques
- embedded diagnostic control mechanisms
- coordinated factory-to-field deployment methodologies

within a unified runtime integrity framework.

This approach enables structural diagnostics to operate under conditions more closely aligned with:

- real workload behavior
- runtime thermal conditions
- voltage and droop behavior
- timing-sensitive execution scenarios
- lifecycle degradation effects
- silent data corruption mechanisms and latent runtime failures

Importantly, VegaTest does not eliminate conventional DFT methodologies. Instead, it establishes an architectural evolution capable of coexisting with existing scan, ATPG, LBIST, MBIST, SLT, and monitoring ecosystems while progressively reducing their infrastructure burden, deployment fragmentation, and runtime limitations.

By combining runtime-aware orchestration with structural diagnostic rigor, VegaTest bridges the gap between deterministic structural validation and real operational behavior.

4.2 Architectural Impact and Deployment Benefits

The VegaTest architectural direction enables several potential advantages across semiconductor deployment flows.

Cost and Infrastructure

- substantial reduction in production test time
- lower ATPG vector requirements, especially for at-speed testing
- reduced ATE dependency and manufacturing cost
- reduced dependence on pervasive scan infrastructure

Test Quality

- improved structural coverage and diagnostic quality

- reduced defect escapes and silent data corruption risk

Lifecycle and Productization

- improved yield and binning efficiency
- improved lifecycle diagnostics and field reliability visibility
- stronger alignment between manufacturing validation and runtime operation
- unified factory, SLT, in-field, SLM, and RAS deployment strategies

Fundamentally, VegaTest shifts semiconductor validation from a manufacturing-centric test activity into a continuous runtime-aware structural integrity framework spanning production, deployment, and operational lifetime. In doing so, it impacts the full economics and quality envelope of silicon validation: diagnostic coverage, test cost, ATE dependency, vector volume, production test time, yield, binning, field reliability, SLM, and RAS.

5 Why This Matters Now

Key Observations

- Advanced nodes, AI workloads, automotive systems, and chiplet architectures are creating failure modes that are increasingly runtime-dependent and difficult to reproduce during conventional manufacturing test.
- Silent data corruption, intermittent faults, thermal hotspots, power-integrity events, and aging effects require stronger structural integrity awareness across the deployed lifetime of silicon.
- Conventional validation cost is scaling aggressively through larger vector volumes, longer ATE time, heavier DFT infrastructure, package/GPIO overhead, SLT expansion, and monitoring complexity.
- The industry is moving toward lifecycle observability and in-field diagnostics, but the next step is connecting runtime awareness with deterministic structural integrity.

Modern semiconductor validation is being pushed by multiple forces at the same time. The challenge is no longer limited to detecting manufacturing defects before shipment. Modern SoCs must preserve correctness, reliability, and structural integrity across dynamic workloads, advanced-node variability, long product lifetimes, and increasingly mission-critical deployment environments.

The pressure is coming from several directions:

Technology Scaling Pressure

- shrinking voltage and timing margins
- increasing variability and interconnect sensitivity
- localized thermal behavior and aging acceleration
- runtime-sensitive failure mechanisms

AI and High-Performance Compute

- bursty workload behavior
- extreme parallel switching activity
- transient droop and hotspot conditions
- long-duration runtime exposure
- growing sensitivity to silent data corruption

Automotive and Mission-Critical Systems

- ASIL and functional safety requirements
- near-zero defect escape expectations

- deterministic correctness requirements
- long operational lifetime demands
- safety exposure to intermittent and latent failures

Validation Cost Explosion

- large ATPG vector growth
- increasing ATE execution time
- pervasive scan and DFT overhead
- package/GPIO scaling pressure
- expensive SLT and runtime infrastructure

Lifecycle and Reliability Challenges

- workload-dependent degradation
- intermittent runtime failures
- field escapes and latent defects
- deployment lifetime integrity
- fleet-scale reliability management

Heterogeneous and Chiplet Systems

- cross-domain interaction complexity
- distributed thermal and timing behavior
- fragmented validation environments
- increasing runtime coordination challenges

Industry Transition Toward Runtime Lifecycle Awareness

- Silicon Lifecycle Management
- runtime telemetry and observability
- predictive maintenance
- in-field diagnostics
- fleet-scale analytics and reliability management

These trends reflect a broad industry transition toward continuous silicon lifecycle awareness. However, observability alone does not guarantee structural correctness. The remaining transition is the convergence of runtime awareness with deterministic structural integrity assessment.

VegaTest is positioned as a framework for that transition.

6 Conclusion

Modern semiconductor systems are increasingly shaped by runtime-dependent behavior, advanced-node variability, heterogeneous integration, AI-driven workloads, and long operational lifetimes. Under these conditions, conventional validation methodologies become increasingly fragmented between manufacturing test, system-level validation, in-field diagnostics, telemetry, and lifecycle monitoring, while the associated infrastructure cost and complexity continue to grow.

This paper introduced VegaTest™ as a unified runtime structural diagnostics architecture designed to bridge the growing gap between deterministic structural validation and real operational behavior. By combining runtime-aware orchestration, execution-coupled validation, coordinated factory-to-field deployment, and stronger lifecycle structural integrity awareness, VegaTest establishes a framework that extends structural diagnostics beyond isolated manufacturing events and toward continuous silicon integrity across production, deployment, and operational lifetime.

Fundamentally, the semiconductor industry is moving toward continuous silicon lifecycle awareness. The next architectural transition is extending that awareness into deterministic runtime structural integrity. VegaTest is positioned as a framework for that transition.