SOLUTION BRIEF

SOS Core: Reliable, High Performance Design Data Management for Semiconductor Engineering Teams

KEYSIGHT

Table of Contents

Executive Overview

The semiconductor industry is experiencing unprecedented growth, with global revenues projected to approach $700 billion in 2025 and the EDA software market expanding at roughly 8.5% CAGR toward $29 billion by 2030. This expansion is driven by the surge in AI accelerator chips, shift-left design practices, heterogeneous integration, and increasingly distributed engineering teams.

These dynamics are creating exponential growth in design data volume and complexity. Engineering organizations that lack structured, reliable design data management face mounting risks: version conflicts, costly re-spins, lost institutional knowledge, and wasted engineering cycles.

Keysight Design Data Management (SOS) Core addresses these challenges directly. Built for small to mid-sized semiconductor design teams, SOS Core delivers enterprise-grade version control, seamless EDA tool integration, metadata-driven IP reuse, and a high-performance architecture optimized for the large binary files characteristic of analog, RF, and mixed-signal design workflows. It provides a reliable, scalable foundation for teams that need performance and simplicity without sacrificing data integrity.

This document details the engineering data management challenges SOS Core solves, explains its core capabilities and architecture, and articulates the value it delivers to semiconductor design organizations.

The Challenge: Design Data Management in Semiconductor Engineering

Design data management is a mission-critical function in any modern semiconductor project. Yet a surprising number of engineering organizations still rely on ad hoc methods: shared network drives, manual file copies, ambiguous naming conventions, and email-based IP transfers. The consequences compound as teams grow, projects multiply, and schedules compress.

The Problem of Unmanaged Design Data

Consider a typical scenario: a team of analog designers works on a project stored on a shared network drive. The drive contains documentation, configuration files, PDK data, and the design objects themselves. Without a data management system, any engineer who modifies a file changes it for everyone, instantly and irreversibly.

To avoid disrupting colleagues, engineers resort to copying project data into personal scratch libraries, local directories where they can work in isolation. The immediate consequence is data proliferation: the same design block now exists in multiple copies across multiple machines, each potentially diverging from the others.

Without automated version tracking, engineers fall back on naming conventions to distinguish iterations: "amplifier_final," "amplifier_final_v2," "amplifier_FINAL_reviewed." This ad hoc

versioning creates ambiguity. No one knows with certainty which version is current, which has been verified, or which was used for a particular tapeout.

The Cost of Uncontrolled Design Data

The costs of this unmanaged approach manifest in several critical ways:

- **Merge conflicts and data loss.** When engineers finish their work in scratch libraries and attempt to merge changes back into the production environment, conflicts are almost inevitable. Two engineers may have modified the same cell, and one version will silently overwrite the other. There is no accountability, no history, and no mechanism to detect or recover from these collisions.

- **Broken traceability.** If a customer reports a defect in a production chip, the team must determine exactly which design versions were used at tapeout. Without systematic versioning and labeling, this is guesswork. Engineers sift through email chains, timestamps, and folder structures trying to reconstruct the design state months or years after the fact.

- **Wasted engineering effort.** Engineers spend hours searching for the correct version of a library cell, debugging issues caused by stale data, or recreating blocks that already exist somewhere in the organization. This is time not spent on design innovation.

- **Release and tapeout risk.** Without a structured release process, tapeout preparation depends on manual data collection: zipping directories, manually checking that the right versions are included, and hoping nothing was missed. Any error at this stage can result in a re-spin costing hundreds of thousands to millions of dollars.

Multi-Site and Scaling Challenges

All of these problems are amplified when teams are distributed across geographies. A design house with analog engineers in Europe, layout teams in Asia, and verification resources in North America faces not only the data integrity challenges above, but also the physics of network latency. Transferring large EDA datasets over wide-area networks is slow. Real-time synchronization is difficult. Teams in remote offices work with stale data, leading to version drift and duplicated effort.

This is the environment SOS Core is designed to address.

The Solution: SOS Core

Product Overview

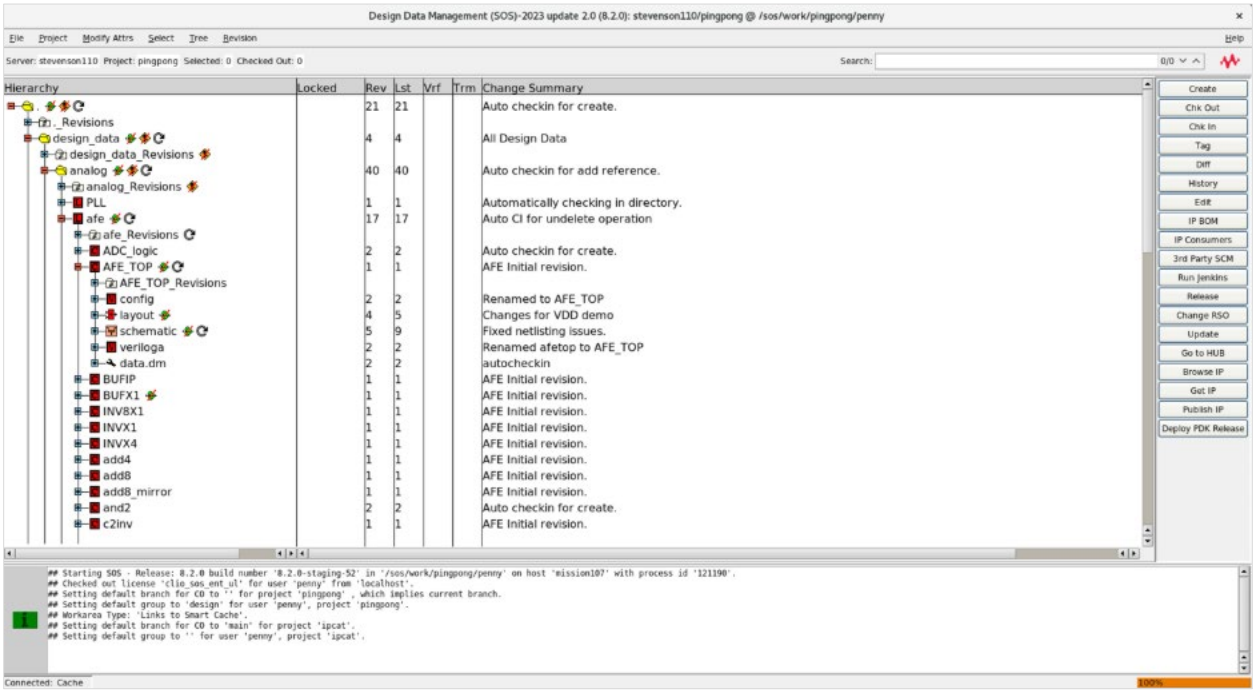

Figure 1. SOS Core is built on reliable and trusted architecture

SOS Core is Keysight's design data management platform built for small to mid-sized engineering teams that need performance, reliability, and simplicity. It provides high-performance version control of design data, seamless collaboration across teams and workflows, and referencing mechanisms for IP reuse-driven design.

With EDA-native integrations and an intuitive setup [figure 1], SOS Core eliminates version conflicts, reduces rework, and accelerates design cycles, empowering teams to focus on innovation instead of data management. SOS Core is built on the same architecture trusted by the world's leading semiconductor companies and offers a clear upgrade path to SOS Enterprise as organizational needs grow.

SOS Core delivers enterprise-grade design data management in a streamlined, easy-to-deploy package. It empowers small to mid-sized design teams with fast, reliable version control, seamless EDA integration, and metadata-driven IP reuse.

Core Capabilities

Version Control and Data Integrity

At its foundation, SOS Core provides automated, granular version control for all design artifacts. Every modification to a managed file creates a new, immutable revision in the central repository. The system maintains a complete history of every design object: who changed it, when, and what was changed.

Crucially, SOS communicates version status to engineers through a clear iconography system. In both the SOS GUI and within supported EDA environments (such as the Cadence Virtuoso Library Manager), status icons appear next to every managed object. A red indicator signals that the local copy is out of date; a green indicator confirms currency. Tooltips provide detail: "This object needs an update because a newer version exists." This eliminates the fundamental ambiguity that plagues unmanaged environments.

Check-in, Check-out, and Locking

SOS Core implements a check-out/check-in workflow with pessimistic locking, the standard model for hardware design data where concurrent modification of binary files is impractical.

When an engineer checks out a design object, two things happen simultaneously: the engineer is granted write access to that object, and a lock is placed on the object in the repository, preventing any other user from modifying it. Other users can still view the object and can see, through the GUI's lock icon [figure 2], exactly who has it checked out. When the engineer is satisfied with the changes, a check-in commits the modifications and creates a new revision. The lock is released, and the updated version becomes available to the rest of the team.

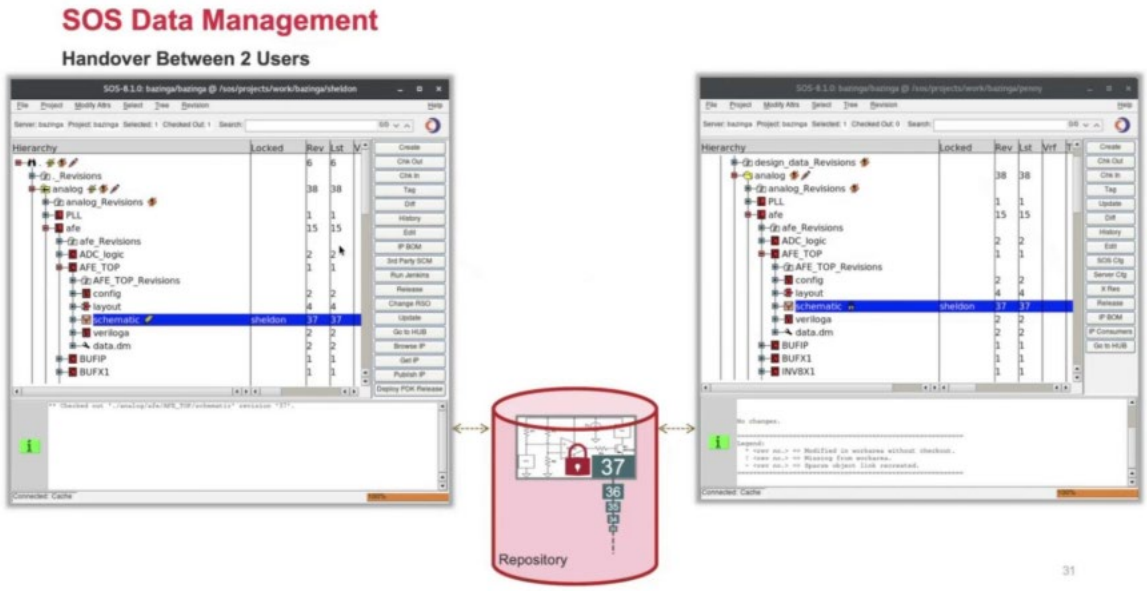

Figure 2. The SOS Core GUI lock icon prevents engineers from unknowingly overwriting colleagues' work

This mechanism guarantees that two engineers never unknowingly overwrite each other's work. It provides both concurrency control and full accountability.

Labels and Revision Search Order (RSO)

SOS Core introduces a powerful abstraction layer on top of raw version numbers through its labeling and Revision Search Order (RSO) system [figure 3].

Labels are human-readable aliases assigned to specific revisions. Instead of remembering that the verified schematic is "revision 34," an engineer tags it "Gold" or "design_done." Labels serve as semantic milestones in the design lifecycle.

The **Revision Search Order (RSO)** is a property of each user workarea that acts as a priority list of labels. When a workarea is populated, SOS resolves which revision of each object to retrieve by walking the RSO from highest to lowest priority. For example, an RSO of [Experimental, Gold, Main] will first look for a revision tagged "Experimental"; if none exists for a given object, it falls back to "Gold"; if that is also absent, it retrieves the latest version (Main).

The practical power of this system is significant. A design engineer can tag a schematic "design_done," and a layout engineer in a different timezone can configure their RSO to automatically pull that exact version without any manual coordination, email exchange, or file transfer. An LVS verification engineer can then set their RSO to [design_done, layout_done] to assemble exactly the right combination of schematic and layout revisions for verification. The entire handoff process is formalized, automated, and traceable.

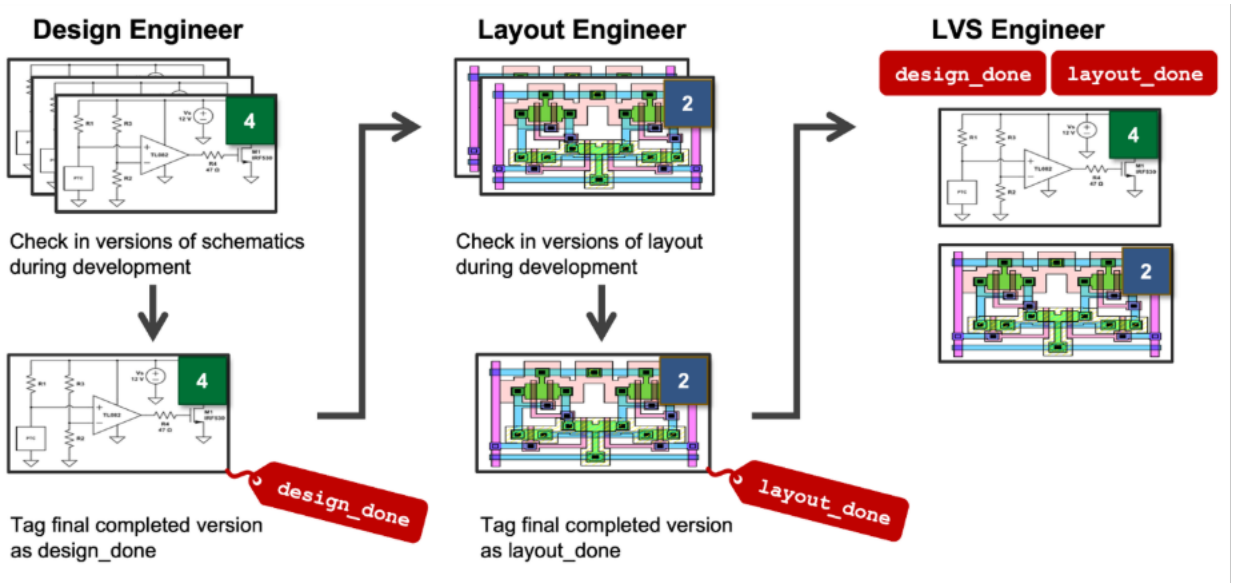

Figure 3. Using Labels and RSO to tag specific milestones

Snapshots

SOS Core supports project snapshots: point-in-time captures of the entire project state. A snapshot freezes the exact combination of revisions across all objects in a project at that moment. If subsequent changes introduce problems, the team can revert the entire project to the snapshot state. Snapshots function as a project-level safety net, providing a reliable baseline for tapeout preparation, milestone reviews, or risk mitigation.

Architecture

SOS Core's architecture is intentionally straightforward, consisting of three principal components: the Primary Repository, Cache Repositories, and User Workareas [figure 4].

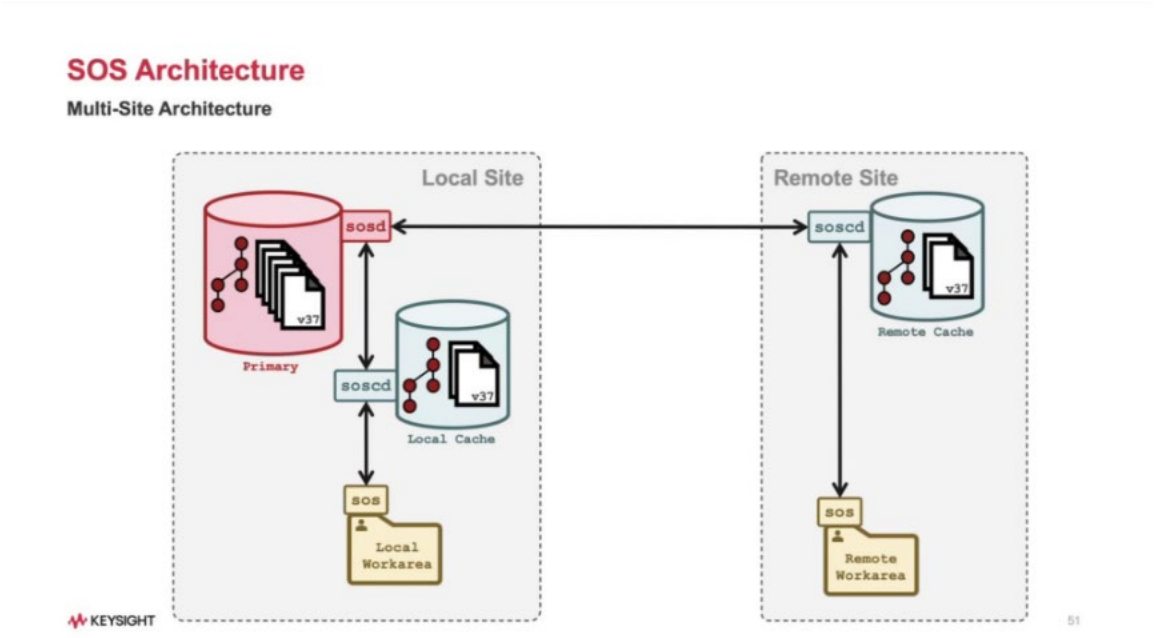

Figure 4. Linking data between local and remote sites using SOS Core's multi-site architecture

Primary Repository

The Primary Repository is the central, authoritative store for all versioned design data and metadata (timestamps, authorship, labels, attributes). It maintains the complete revision history for every managed object. All projects are organized within the repository as logical partitions.

Cache Repositories

Cache Repositories act as local mirrors, placed at each site where engineers work. Rather than connecting directly to the Primary (which may be geographically distant), users connect to their local Cache. The Cache synchronizes with the Primary using optimized algorithms, ensuring that local users experience local-speed data operations regardless of where the Primary is hosted.

Caches serve a second, equally important role: they enable SOS Core's advanced workspace technologies, Links-to-Cache and Sparse Populate, which are described below.

Links-to-Cache Technology

This is one of SOS Core's most distinctive performance differentiators. In conventional version control systems, each user's workspace contains a full physical copy of the project data. For a project with 100 GB of design data and 50 engineers, this means 5 TB of duplicated storage consumed by workspaces alone.

SOS Core's Links-to-Cache technology takes a fundamentally different approach [figure 5]. User workareas contain symbolic links that point to the single copy of each file stored in the local Cache. No data is duplicated. A workarea for a 100 GB project may consume only a few megabytes. When an engineer opens a file through their EDA tool, the symbolic link resolves transparently to the cached copy.

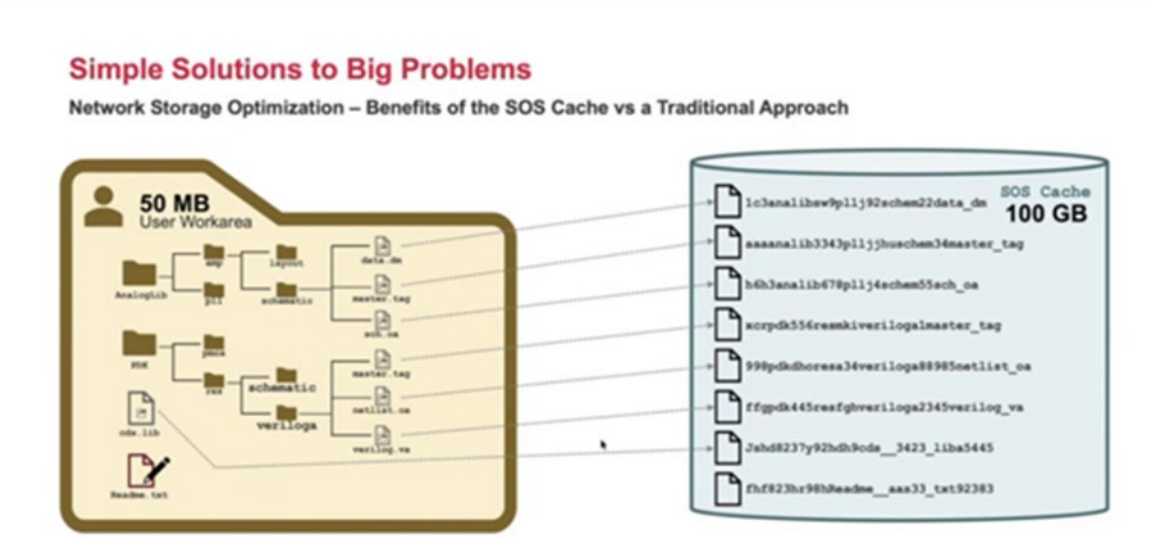

Figure 5. When a user checks out a file, the symbolic link is replaced by an actual copy the user can edit. When the user checks in the file, the changes are committed to the cache, and the copy once more replaced by the link.

At enterprise scale with hundreds of engineers, this approach delivers dramatic storage savings and eliminates the time-consuming workspace population step that characterizes competing solutions.

Sparse Populate

Sparse Populate extends Links-to-Cache for projects of extreme scale. In projects containing very large subsystems, such as PDKs with tens of thousands of files, the number of individual symbolic links can approach operating system limits. Sparse Populate addresses this by creating a single symbolic link to a top-level directory, with the directory structure and contents hosted entirely within the Cache [figure 6]. This reduces the symbolic link count by orders of magnitude while preserving the same lightweight workspace behavior.

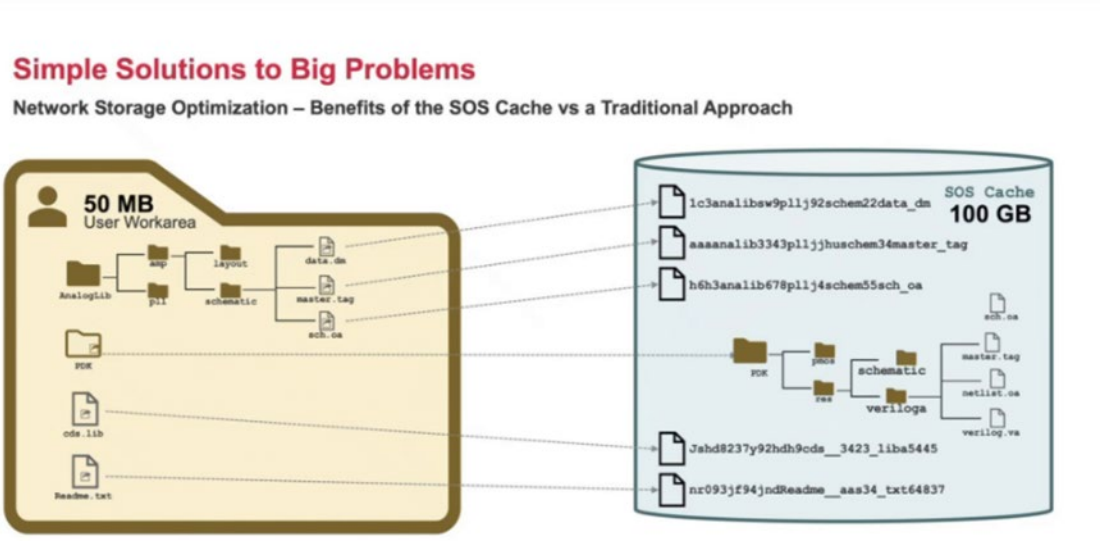

Figure 6. Sparse Populate- 1 single link. The tree is built on the cache

Security and Data Protection

SOS Core provides multiple layers of data protection:

- **Access Control Lists (ACLs).** Administrators define groups, assign members, and set per-object permissions. Critical design files can be restricted to specific teams or individuals, ensuring that only authorized engineers can modify sensitive IP.
- **Server-side authentication.** Users authenticate with credentials when connecting to SOS, adding an identity verification layer beyond filesystem permissions.
- **SSL encryption.** For multi-site deployments where data traverses the public internet, SSL encryption protects data in transit between Caches and the Primary Repository.
- **Project Views.** Project Views allow administrators to define filtered perspectives of a project. For example, analog engineers can be restricted to seeing only analog data within a mixed-signal project, while digital engineers see only their domain. Data remains in a single project but is logically segmented for security and focus.

Backup, Recovery, and High Availability

SOS Core supports two backup mechanisms:

- **Hot Backup.** An always-on, background process that continuously replicates repository data to a designated backup repository. If the production repository is compromised, the data can be recovered from the backup. This operates transparently to users.
- **Project Archive.** An on-demand mechanism that exports a specific project as a standalone archive. Useful for milestone preservation or selective backup of individual projects.

For teams requiring zero-downtime guarantees, SOS Core also supports replication. A Replica Repository mirrors the Primary in real time. If the Primary becomes unavailable, the Replica can be promoted to become the new Primary with a single operation. All connected Caches and workareas fail over automatically, with no disruption to active users. This capability also supports planned maintenance windows: promote the Replica, perform maintenance on the original Primary, and revert when complete.

IP Reuse and Knowledge Management

A significant source of wasted effort in semiconductor design is the repeated recreation of existing IP blocks. A PLL designed for one project, a bias circuit validated for another, a standard cell library qualified on a given process node: all of this work has value beyond its original context. Yet without a structured reuse mechanism, IP is typically shared by informal means, usually by copying files between engineers or project directories, losing all provenance and version history in the process.

SOS Core addresses this through its references mechanism. Rather than copying IP from one project into another (which duplicates data and breaks traceability), SOS Core creates a reference: a managed link from the consuming project to a specific version of the IP in its source project. The referenced IP appears in the consuming project's workspace as if it were local, but it is not duplicated. Its provenance is fully tracked. If the source IP is updated, the consuming project can choose to pull the update or remain on the pinned version.

This transforms IP reuse from an ad hoc, error-prone practice into a governed, traceable workflow. Engineering organizations can build libraries of validated, reusable IP blocks and share them across projects with confidence in their integrity and lineage.

Automation with SOS Triggers

SOS Core supports event-driven automation through its Triggers framework. Triggers are hooks that execute custom scripts or functions when SOS commands are invoked. They come in two types:

- **Pre-event triggers** execute before the SOS command and can conditionally allow or block the operation based on custom logic. Examples include: blocking a Verilog check-in unless the file passes a lint check, requiring a description to accompany every check-in, or preventing accidental check-in of files exceeding a size threshold.

- **Post-event triggers** execute after the SOS command completes and are used for downstream automation. Examples include: sending notification emails after check-in, generating reports, updating dashboards, or triggering CI/CD pipelines.

Triggers allow teams to encode their specific process requirements directly into the data management workflow, reducing human error and enforcing consistency without manual oversight.

EDA Tool Integrations

SOS Core's value is realized most directly through its integrations with the design environments engineers use daily. SOS maintains the broadest portfolio of EDA integrations in the design data management category:

EDA Vendor	Supported Tools
Cadence	Virtuoso Studio (includes Visual Design Diff/VDD)
Synopsys	Custom Compiler, Opto Compiler, Laker
Siemens EDA	Pyxis, Tanner
Keysight	Advanced Design System (ADS)
Silvaco	Gateway + Expert
MathsWorks	MATLAB
Empyrean	Aether

These integrations are not superficial file-system bridges. They are EDA-native: they understand the library, cell, and cellview hierarchy of each tool. Version status icons appear directly in the Library Manager. Check-in, check-out, update, and diff operations are accessible from within the design environment's context menus. Engineers interact with SOS through their familiar EDA interface, not a separate application [figure 7].

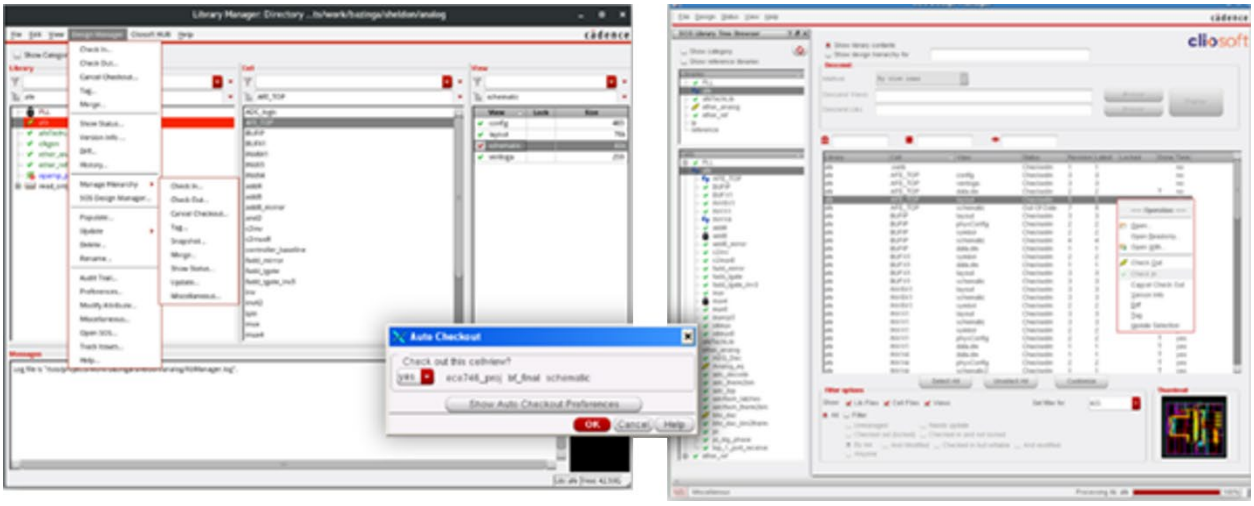

Figure 7. SOS Core seamlessly integrates with many of the major EDA tools

A standout capability within the Cadence Virtuoso integration is the Visual Design Diff (VDD) tool. VDD enables side-by-side visual comparison of two cellview revisions (or two entirely different cellviews) [figure 8]. Differences are highlighted with color-coded halos directly on the schematic or layout canvas, and an exhaustive differences browser lists every detected change. Non-functional differences (e.g., cosmetic wire repositioning with no circuit impact) can be filtered out. VDD supports hierarchical comparison across design hierarchies. This capability is valued so highly that some customers adopt it as a standalone tool independent of the broader SOS platform.

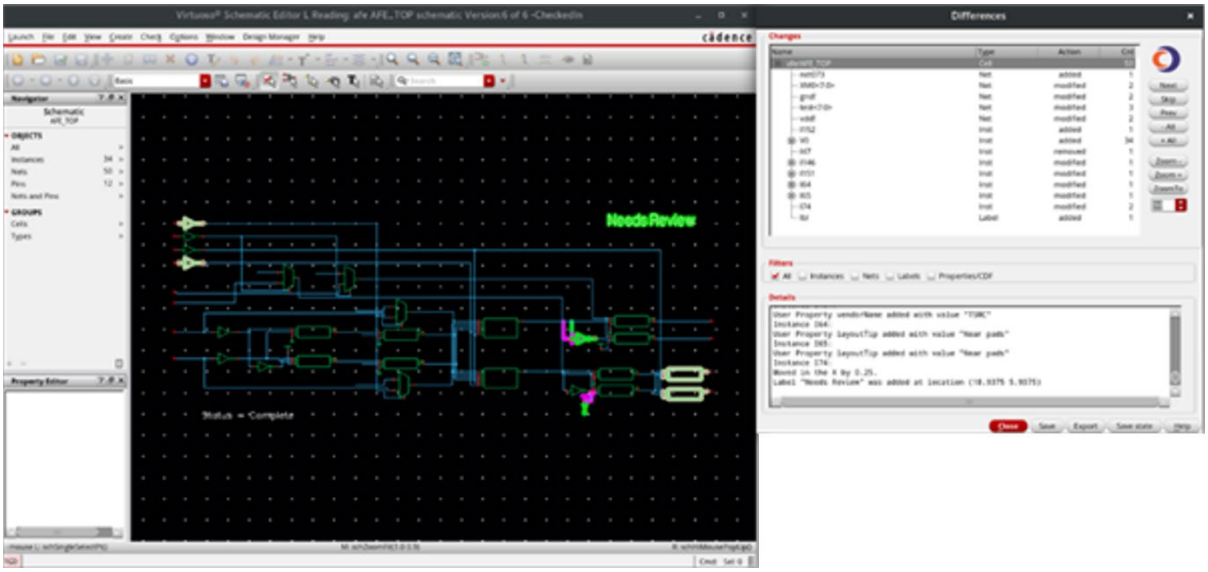

Figure 8. On-GUI comparison of two cellviews using Visual Design Diff (VDD)

Why SOS Core works

SOS Core delivers measurable value to semiconductor design organizations across four principal dimensions:

Fast, Reliable Version Control

Every engineer works on the correct version with full confidence. Version conflicts are eliminated by design through the lock-based check-in/check-out model. The complete revision history of every object is preserved and searchable. Engineers no longer waste time asking "which version is the latest?" or debugging issues caused by stale data.

Seamless Collaboration

SOS Core enables concurrent work by multiple engineers on the same project without the friction and risk of manual data transfers. The Cache architecture ensures that remote teams experience local-speed performance. Labels and RSO automate design handoffs between functional roles (design, layout, verification) without manual coordination. The entire team operates from a single source of truth.

Knowledge and IP Reuse

SOS Core transforms past work into future value. Its references mechanism provides traceable, non-duplicative IP sharing across projects. Metadata and lineage are automatically captured, making it straightforward to find, evaluate, and reuse validated design blocks. Organizations build a compounding IP asset base rather than repeatedly recreating existing work.

Scalable by Design

SOS Core is designed to grow with the organization. Links-to-Cache and Sparse Populate keep workspace overhead minimal regardless of project size. The Cache architecture supports multi-site deployment. And when the organization’s needs evolve to require full lifecycle traceability, governance automation, or AI-ready data infrastructure, SOS Core provides a seamless upgrade path to SOS Enterprise with no data migration required.

Value Proposition Summary: *SOS Core empowers small and mid-sized design teams to work efficiently, avoid data loss or conflicts, and accelerate reuse, all within a simple, scalable foundation that can later expand to enterprise deployment.*

Target Users and Key Applications

Organization Segments

SOS Core is purpose-built for:

- Small to mid-sized semiconductor companies
- Design service providers and IP houses
- Academic and research institutions
- Single-site or departmental teams within larger corporations focusing on specific projects or product lines

Workflow Domains

Workflow Domain	Typical Activities	SOS Core Value
Analog/Mixed Signal IC Design	Schematic capture, simulation (Spectre, HSPICE), Monte Carlo/corner analysis, LVS/DRC, PDK management	Version-accurate design management, schematic-layout consistency, metadata-driven IP reuse
RF and High Frequency Design	RF schematic and EM simulation (ADS, EMX), parasitic modeling, sim-to-measurement correlation	Large EM dataset handling, cross-domain data linking, ADS-native integration
Custom Digital/Mixed-Signal	Digital control block integration, power domain verification, SDL, mixed-signal LVS	Version control of configurations, dependency tracking across analog and digital domains
Physical Implementation	Layout editing, DRC/LVS, parasitic extraction, tapeout prep	Revision tracking, schematic-layout alignment, change detection between layout versions
Verification/Simulation	Regression setup, netlist/waveform management, version-to-simulation correlation	Design-to-simulation lineage, reproducible regression results, large dataset management

Who it's built for

- **IC Design Engineers** (analog, digital, mixed-signal) who need fast, conflict-free access to design data
- **RF Engineers** managing large EM simulation datasets and requiring traceability between design and measurement
- **Layout Engineers** who need schematic-layout version alignment and rapid change detection
- **Verification Engineers** who require traceable links between design revisions and simulation results
- **CAD/EDA Managers** responsible for tool infrastructure, workflow standardization, and data governance
- **Engineering Directors** (buyers) seeking to improve team productivity, reduce re-spins, and control design data risk

The Competitive Advantage

SOS Core operates in a market alongside general-purpose version control systems, EDA-oriented data management tools, and homegrown solutions. SOS Core differentiates on several dimensions:

Differentiator	SOS Core Advantage
Links-to-Cache workspaces	User workareas are lightweight symbolic links, not full data copies. Competing solutions duplicate data into workspaces, consuming storage proportional to *project size × number of users*. SOS Core workspaces remain in the megabyte range regardless of project scale
EDA-native integrations	The broadest and deepest set of EDA tool integrations in the category, with mature support for Virtuoso, Custom Compiler, ADS, Pyxis, Tanner, and more. Includes the Visual Design Diff (VDD) tool for Virtuoso
Analog/RF design heritage	SOS was purpose-built for the large binary files and library-cell-cellview hierarchies of analog and RF design. Competitors originate from software/digital design contexts where performance constraints differ.
Ease of use and deployment	Repeated customer feedback cites SOS as having the shallowest learning curve and simplest administration in its category. It scales to enterprise environments without requiring dedicated IT staff for ongoing management.
Lower total cost of ownership	Compared to open-source/homegrown solutions, SOS delivers lower TCO when accounting for the engineering time consumed by maintenance, custom integration development, and the productivity loss from lacking dedicated support.

Scaling your Approach

SOS Core is designed as the entry point to the broader Keysight Design Data Management platform. As organizations grow in team size, number of sites, regulatory requirements, and data maturity, SOS Core provides a seamless upgrade path to **SOS Enterprise**, which extends the platform with:

- Global, multi-site collaboration with real-time synchronization and unified project visibility
- Enterprise-grade governance: role-based access control, encryption, automated approval workflows, and end-to-end audit trails
- Full lifecycle traceability from schematic through verification and test
- AI-ready data infrastructure with structured metadata and APIs for ML pipelines and agentic AI workflows
- IP lifecycle management with catalogs, bills of materials, and standardized reuse processes

The upgrade preserves all existing project data, configurations, and user workflows. There is no migration penalty.

Conclusion

Design data management is no longer optional in semiconductor engineering. The cost of unmanaged data, measured in re-spins, lost IP, duplicated effort, and compliance exposure, is too high for any organization competing in today's market.

SOS Core provides the solution that small to mid-sized design teams need: fast, reliable version control with EDA-native integration; lightweight, high-performance workspaces that scale without storage penalties; traceable IP reuse that compounds organizational knowledge; and a simple, proven architecture that can grow to enterprise scale.

Built on the same platform trusted by the world's leading semiconductor companies, SOS Core is the foundation for design data management in the AI era.

To learn more, visit: SOS Core | Keysight