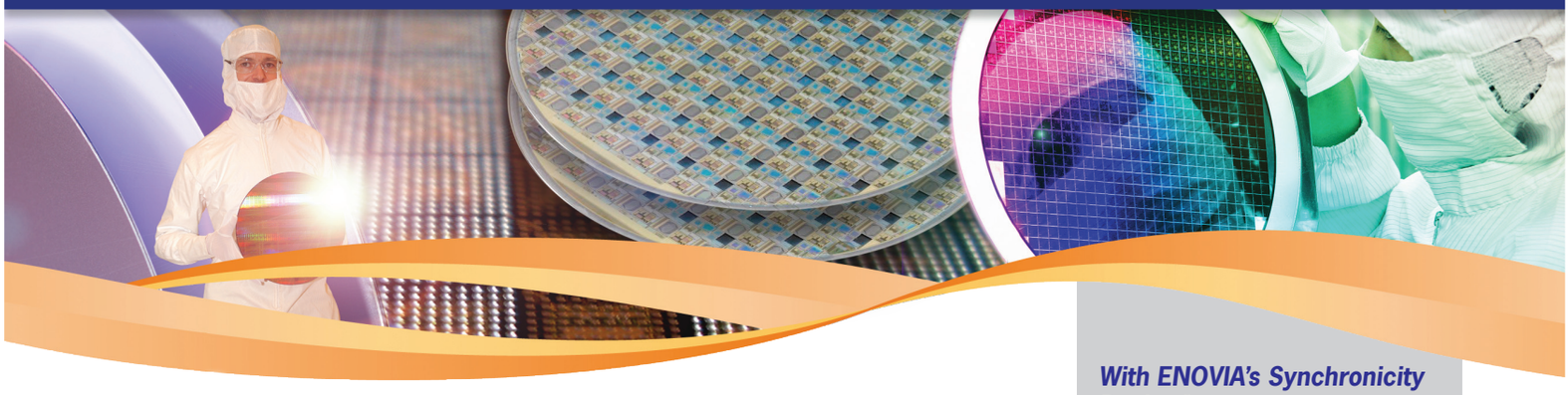


# ENOVIA Synchronicity
## DesignSync®

The creation of complex electronic products is not an easy proposition, and is becoming increasingly complex with the proliferation of globally dispersed teams. Since 1998, integrated circuit design (IC) teams have relied on ENOVIA's Design Data Management (DDM) solution, Synchronicity DesignSync, to help manage the hardware and software data in their products. Today, over 120 development organizations, including 13 of the top 15 semiconductor companies, take advantage of DesignSync to boost design productivity.

***Today's reality — design teams, and the DDM tools they use, are increasingly distributed and specialized.***

Designing an IC today requires integrating many and various datasets that are contributed to multiple, and often geographically dispersed, design teams. As ICs become more complex, design teams are required to specialize because no one individual, or even a team of individuals, can do it all. Specialization results in the segregation of design tasks along specific lines such as analog or digital design – and then within those teams, further segregation occurs. For example, in analog design, schematic capture and physical layout are often performed by different design teams. In digital design, specialties exist in the areas of RTL design, design compilation, physical place and route, simulation and verification. ICs also include ever-increasing amounts of embedded software, which is contributed by yet another team.

All components contributed by the different design teams need to come together into one chip design, a task that is managed by an integration team. While individual teams are focused on the details of the design data for which they are responsible, and are thus focused at the file level, the integration team needs to be able to manipulate data sets at a higher level of abstraction. Glue logic at the top-level instantiates configurations, or releases, of lower-level modules. This sounds straightforward, but in many cases it is not because, more often that not, individual design teams are using independent and unconnected DDM systems that require that the integration team extract data sets for different component parts from different systems. While the various SCM (software configuration management) and DDM tools in use by the individual design teams may satisfy local requirements, design data management breaks down at the integration level. As a result, manual procedures are used, leading to errors and inefficiencies. For this reason, many companies are now attempting to plot a roadmap that leads from a policy of local optimization into the realm of global design efficiency. DesignSync is one solution that can help.

### *With ENOVIA's Synchronicity DesignSync you can:*

- **Use your design chain as a competitive weapon**
- **Connect and manage your entire design chain with a unified Design Data Management system**
- **Significantly boost design productivity for a rapid payback and strong ROI**
- **Maximize your ability to reuse existing designs and embedded software**
- **Manage your design hierarchy as part of the design process**
- **Reduce time-to-market by increasing collaboration efficiency**
- **Win the first-to-market advantage**
- **Manage complex data types from a variety of EDA tool vendors.**

DASSAULT SYSTEMES

## Designing with Modules

DesignSync enables design teams to manage data at both the detailed file/directory level, and at a "modular" level of abstraction. It is the ability to efficiently manage design data at both of these levels of abstraction which differentiates DesignSync from other DDM systems. Design data contributed by individual teams can be seamlessly integrated into higher level designs.

At the heart of the architecture is the "module." A module represents a single coherent and consistent collection of files and folders. While the revision history of individual files is maintained at the file level, a revision history of the context, or structure, in which the files exist is maintained at the "module" level. A module version is stored in a "manifest" which represents a "bill of materials" for the module version. The manifest stores a list of individual file versions along with the associated directory structure, and, if hierarchy is included, references to sub-modules. The revision history of a module records the complete genealogy, making it possible to resurrect the design as it existed at any previous point in time.

The architecture under which module versions are stored in a DesignSync server affords many advantages:

- ***Change Set Processing***
  Because "manifests" are stored for each module version, update operations employ a "change-set" paradigm, eliminating the need to enumerate over all objects in a directory structure to determine which subset needs updating.

- ***High Performance***
  Change set processing also enables high performance, and high performance substantially increases the probability of the successful deployment of a DDM tool which in turn encourages the use of best practices.

- ***Atomic Operation***
  Operations such as checkin are "atomic" meaning that transactions involving multiple files are not committed to the data repository unless all the individual files have been checked in successfully. Sophisticated recovery mechanisms allow an interrupted checkin to proceed where it left off, without having to start over.

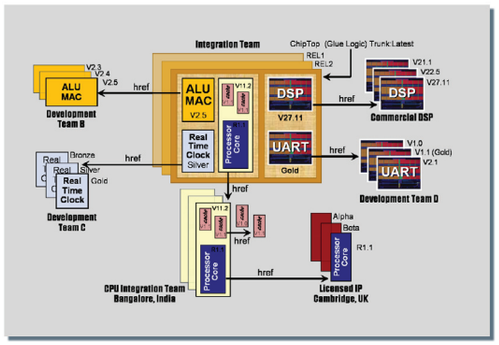
- ***Directory Versioning***
  Creation, deletion or renaming of directories and files is captured in the module version history, providing a complete genealogy of changes to the design.

- ***Distributed Data Repositories***
  Design data can be distributed over multiple repositories, thus maximizing local efficiencies.

## Managing a Design Hierarchy

The real, and differentiating, power of DesignSync is manifested in its patent-pending ability to manage a design hierarchy of modules. In addition to file versions organized in folders, a module version may contain one or more "hierarchical references" (hrefs) to other module versions that may be stored in the same server or in another server located anywhere across the globe. Hrefs are processed when design data is fetched into a workspace. If a module version that contains an href is fetched, the href is followed and the referenced module version is fetched as well. In this manner, a hierarchical design is assembled in the workspace.

By establishing containment relationships (hrefs) to other modules in modules themselves, DesignSync captures the design "hierarchy" and actively manages it throughout the evolution of the design. Thus, all of the individual design elements are managed, and interrelationships are maintained in a consistent manner, whether the entire dataset is stored in a single server, or distributed across servers around the world. In other words, an advantage to using DesignSync is that an entire design hierarchy that is distributed across multiple servers can be fetched with a single command.




Without using a DDM tool that can efficiently manage a hierarchy of constituent components in a complex IC, many companies are struggling with capturing the "hierarchy" of a design even after it has been completed. If the components come from disparate and unconnected DDM systems, it can be very difficult to determine which components even ended up in a given design release.

Because DesignSync allows users to manage a design hierarchy, a company gains advantages including:

- ***Task-based Workspace Creation***
  Workspaces can be tailored to particular tasks such as verification of an immutably-released design hierarchy, a hierarchy containing work in progress, or a mixture of both.

- ***Single Command Builds or Updates an Entire Hierarchy***
  There is no need to fetch individual components of a design; DesignSync will automatically fetch an entire hierarchy.

- ***Overlapping Module Data***
  A module hierarchy can be created to allow data from different modules to be fetched into the same directory in a workspace.

- ***Filters***
  Sophisticated filtering can be applied to files, directories or hrefs to provide fine-grained control over what data is fetched or operated upon. Filters allow designers to construct workspaces comprised of consistent subsets of data, eliminating the need to fetch large amounts of data unnecessary to tasks at hand.

## Submit, Integrate, Test and Release — SITaR

DesignSync provides a well-defined workflow called SITaR – Submit, Integrate, Test and Release. SITaR is a set of commands that leverage the power of module-based design in an environment consisting of multiple design modules that are then aggregated by an integrator into a higher-level system. Though individual design blocks (modules) may be contributed by different teams, design at the block level cannot occur in a vacuum. Simulations must include interactions with the other blocks in the design.

SITaR is based on the notion that there are two fundamental "roles" in play in for module-based design. A "Designer" contributes at the block (or module) level. An "Integrator" is responsible for integrating blocks together into a top-level design, testing the system and releasing the stable "baseline" (a system-level configuration of blocks) from which all subsequent block-level development occurs.

Working within such a flow, a Designer would fetch the current stable baseline into a workspace. The block the Designer is working on would then be put into "edit" mode, and design activities proceed. All simulation takes place in the context of the baseline consisting of all the other blocks.

The key here is that design work and simulation is not done with work-in-progress configurations of other blocks, but only in the context of the stable baseline. When work is complete, the Designer "submits" his module for possible integration into a newer baseline. The Integrator monitors the submission queue – which could contain submissions for multiple blocks – and is able to build a workspace with any mixture of submitted blocks; this is the "Integrate" step. Regression tests are performed against the newly integrated set of blocks (the "Test" step), and if deemed stable by the Integrator, can be released as a new stable baseline (the "Release" step). Designer workspaces could then be updated, fetching the baseline for all blocks not currently in editing.

Thus, all design work at the block level is performed in the context of a stable baseline of the rest of the blocks in the design. All this activity is performed using simple and intuitive commands such as "sitr submit" or "sitr integrate," alleviating the need to educate the contributing teams in the use of the more fundamental module design command set, along with avoiding complicated handoff and tagging schemes. SITaR commands wrap the underlying module commands, such that the power of the module-based DDM architecture is leveraged in the context of this well- defined use model.

## A Unified DDM System is a Major Competitive Advantage

The majority of data management problems associated with integrating large designs can be eliminated if all the data is managed in a single Unified Design Data Management System, which could exist at a single design center, or could be distributed around the world. DesignSync is the DDM system of choice to make such a vision a reality. In 13 of the top 15 global semiconductor companies and in 100s of other organizations, DesignSync is the standard for management of complex EDA data created by hardware design tools from companies such as Cadence, Synopsys and Mentor Graphics. But, DesignSync is also uniquely suited for the management of RTL Verilog or VHDL design, embedded software design, cell library development or even documentation development. A plug-in for the Microsoft Visual Studio IDE (Integrated Development Environment) is included with ENOVIA Synchronicity DesignSync for use by software designers.

DesignSync enables individual design teams to independently release IP modules, while the integration of multiple modules can be managed at a higher level of abstraction. A single command can fetch an entire design hierarchy, and another single command can create an immutable release of the same hierarchy. Inefficient and error prone manual integration procedures can be completely eliminated. Imagine what that would mean in practice at your company!

## Features & Capabilities

- ***Client/Server Architecture***
  The architecture is uniquely suited to support geographically dispersed design teams.
- ***Multisite Version Control***
  A "single source of the truth" is maintained, and made available to designers regardless of their physical location.
- ***Distributed Data Storage***
  Data repositories (SyncServers) can be configured at any design site for maximum local efficiencies. Data from multiple servers can be automatically aggregated in a workspace.
- ***Data Replication***
  Sophisticated caching mechanisms support data replication and minimize disk space usage by creating workspaces using symbolic links to shared read-only data files.
- ***Security***
  Whether transferring data internally or externally, security is ensured using commercial grade 128 bit SSL encryption.
- ***Internet Based Transfer Protocols***
  Data transactions use standard internet protocols and work seamlessly with existing firewalls.

- ***Audit Trails***
  Detailed revision control activity is captured in a database which may be queried using a standard web browser.

- ***Sophisticated Access Controls***
  Protection of valuable design data is ensured by a configurable Access Control System. Data access is controlled in the server, and does not rely on Unix permissions.

- ***Sophisticated Workspace Management***
  Multiple methods are provided for the creation and maintenance of designer workspaces supporting differing work styles. For example, workspaces can be configured to incorporate changes made by others either on demand, or automatically.

- ***Configurable Use Models***
  Both the "Locking" model and the "Non-Locking" model are supported to suit either design team or individual preferences.

- ***Version History Reporting***
  Brief or detailed reports of the version history of an entire module, or an individual design object, provide a complete genealogical record.

- ***Diff and Merge Tools***
  Both graphical and command line diff and merge tools are provided, including TkDiff.

- ***Comparison Utilities***
  Sophisticated utilities allow for the comparison of module versions, releases, hierarchies and workspaces. A workspace can be compared with a known configuration, or even with another workspace.

- ***Graphical User Interface***
  The DesignSync GUI allows for the navigation and manipulation of data at both the detailed file level and at the more abstract module level. Comparison utilities, diff tools and a command line interface are included in the GUI.

- ***Command Line Interfaces***
  Two command line interfaces are provided. The "dssc" shell runs DesignSync commands. The "stclc" shell is a Tcl interpreter into which the DesignSync commands have been linked, providing programatic capabilities and access to other utilities.

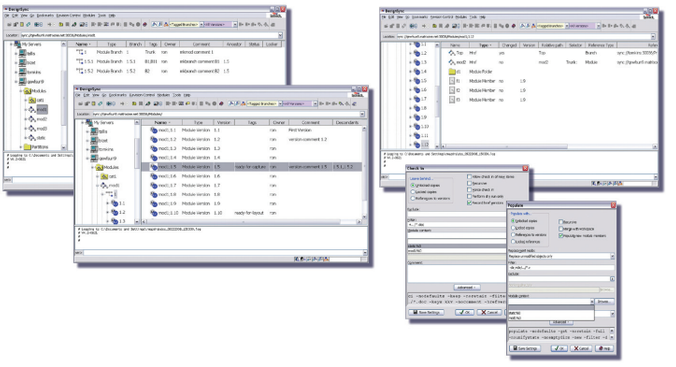
- ***Extensible Architecture***
  The command set can be easily extended by creating aliases or autoloaded Tcl procedures.

- ***Client Side Triggers***
  You can easily introduce process automation to increase efficiency and decrease errors by creating Tcl procedures which are registered to intercept operations and perform other operations. For example, if a layout object is checked in, a Design Rule Check procedure could be automatically run, and if clean, the checkin operation is allowed to proceed.

- ***C-API***
  DesignSync can be easily integrated with other tools using a fully documented C-API.

- ***Plug-in for Source Code Control for Software Components***
  ENOVIA Synchronicity DesignSync includes a plug-in for the Microsoft Visual Studio IDE (Integrated Development Environment.)

- ***"Whereused" Command***
  Given a module version, HCM release, or tagged configuration of files, the "whereused" commmand will trace hrefs upwards, reporting all design hierarchies in which the block has been included. Because design reuse is common, a single component might find its way into many other designs or finished products. By leveraging this capability to extract the answers automatically, countless hours can be saved determining where a given design block has been used.



- ***System Requirements:***
  Operating Systems: Solaris 8,9,10; HPUX 11.11; Linux RH 3,4; Linux SUSE Enterprise 9; Windows XP SP2
  Browsers: Firefox 1.5,2.0; Mozilla 1.7; SeaMonkey 1.0 ; IE 6,7

For additional information, contact us at:
Dassault Systèmes ENOVIA Corp.
900 Chelmsford Street, Lowell, Massachusetts 01851
978 442 2500 • ENOVIA.com • 3DS.com

DASSAULT SYSTEMES

www.3ds.com