

Custom Compiler

Best-in-Class Technology for Custom Design

Accelerating Robust Custom Design

Overview

Custom Compiler™ is a fresh, modern solution for full-custom analog, custom digital, and mixed-signal integrated circuit (IC) design. As the heart of the Synopsys Custom Design Platform, Custom Compiler provides design entry, simulation management and analysis, and custom layout editing features. Designed to handle the most challenging requirements of FinFET process technologies, it delivers industry-leading productivity, performance, and ease-of-use while remaining easy to adopt for users of legacy tools.

Design Entry and Simulation

Custom Compiler provides a highly productive environment for design entry and simulation, with strong features for mixed-signal design, debug, simulation management, analysis, and reporting. Schematic entry has been streamlined with automatic wiring, symbol generation, on-canvas editing of parameters, and more. The Custom Compiler schematic editor includes helpful debugging aids such as a power domain analyzer for checking power supplies and inherited connections across the design hierarchy and a hierarchical net tracer for tracing signals across the design. It also includes a design comparison feature that reports changes between schematics.

Text and schematic views can be freely combined for mixed-signal designs—a language-sensitive editor for Verilog is included. Navigation, cross-probing, and back-annotation of simulation results are available for both schematics and text views. Custom Compiler also has strong features for mixed-signal partitioning and visualization, such as parameterizable interface elements.

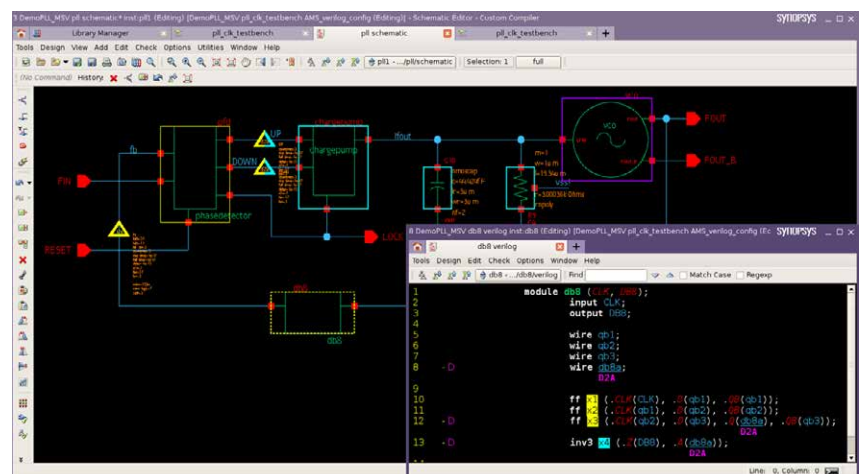


Figure 1: Custom Compiler mixed-signal design entry and debug

Custom Compiler is integrated with Synopsys' PrimeSim™ HSPICE®, PrimeSim SPICE, PrimeSim Pro, and PrimeSim XA circuit simulators and PrimeSim Reliability Analysis technologies. Integrations with many third-party simulators are also available. Simulation support includes PrimeWave™ Design Environment, a comprehensive and flexible environment for setup, run and debug for corners, multi-testbench sweeps, Monte Carlo, Aging/EMIR, and other analyses. PrimeWave Design Environment features a powerful TCL-based scripting capability enabling customization of simulation campaigns including grid-based job distribution and monitoring.

PrimeWave Design Environment has a built-in high-capacity waveform viewer that supports advanced visualization and charting capabilities including statistical analysis, histograms, and scatterplots. It also provides HTML-based reporting to facilitate design reviews.

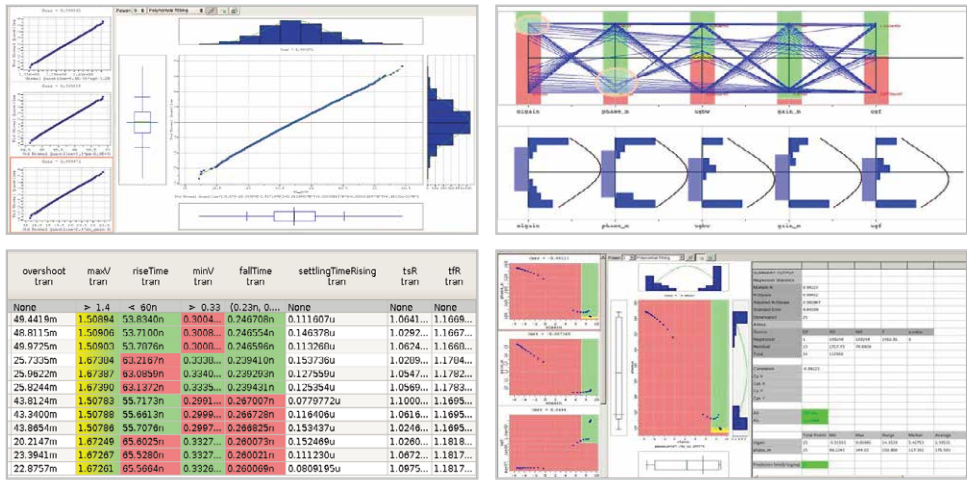


Figure 2: Charting, reporting, and statistical analysis

Custom Layout Editing with Visually Assisted Automation

Custom Compiler includes fast and user-friendly versions of the familiar polygon editing features that users need and then amps up layout team productivity with its pioneering visually-assisted automation (VAA) flow. VAA is an innovative approach to reducing layout effort that has been proven to deliver 2-10X better productivity—especially for difficult FinFET-based designs. VAA leverages the graphical use model familiar to layout designers to provide automation without requiring manual constraint entry. With Custom Compiler, routine and repetitive tasks are dealt with automatically without extra setup.

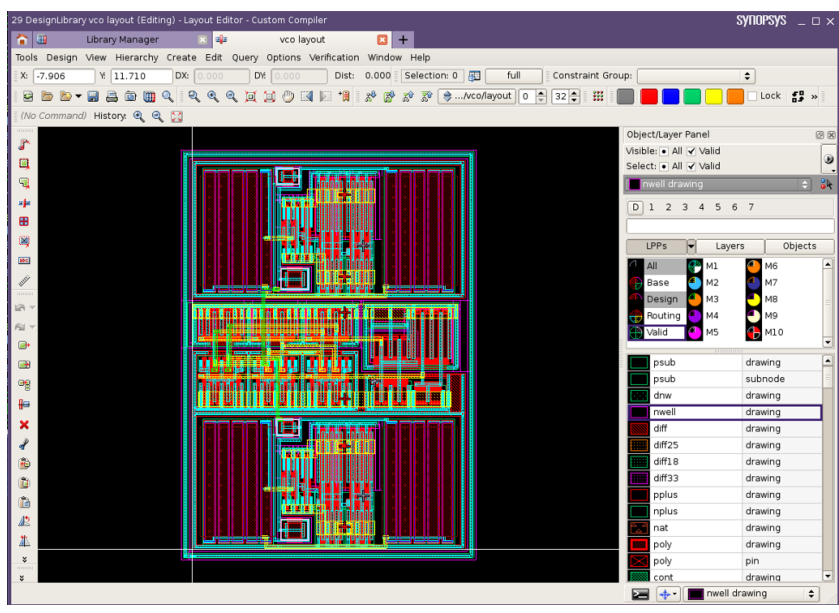


Figure 3: Fast and user-friendly custom layout editing

The Custom Compiler schematic-driven layout capabilities are the foundation of the VAA flow. Schematic-driven layout reduces layout time by checking for connectivity and device parameter mismatches during layout. The Custom Compiler schematic-driven layout has several advanced features. For example, the Custom Compiler integrated symbolic editor lets layout designers place devices into patterns—without worrying about design rule details. This feature is especially useful for creating a matched analog layout or custom digital cells. Custom Compiler includes a pattern router for completing device-level connections.

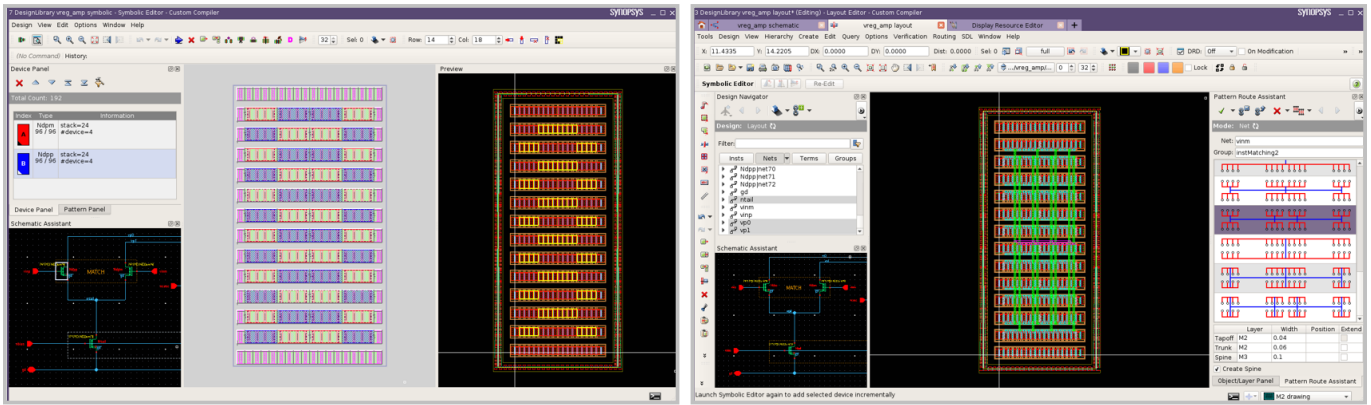


Figure 4: Advanced schematic-driven layout features: symbolic editor (L) and pattern router (R)

A key element of the Custom Compiler VAA flow is the support for layout templates. Layout templates are a way for designers to save and reuse placement and routing patterns. These templates can be applied to generate a layout for similar circuits—even if they don't have the same transistor sizes. Custom Compiler searches the user's template library for templates that match a circuit and generates a DRC-correct layout based on the selected template automatically.

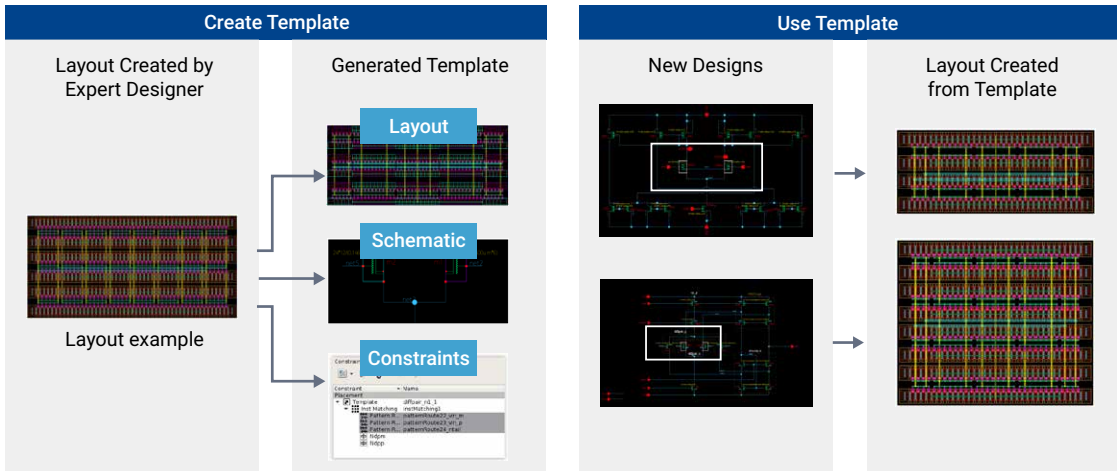


Figure 5: Custom Compiler's user-definable templates enable rapid reuse of design know-how

Custom Compiler includes built-in layout verification features that reduce costly design iterations by catching physical and electrical errors during layout—before signoff verification. These include a design rule checking engine that is fast enough to perform checks dynamically during editing. Other built-in verification features include electromigration checking, and resistance and capacitance extraction. These are available in Custom Compiler for layout designers to check their work as they draw their layout. Unlike other “electrically aware” tools, Custom Compiler's capacitance extraction uses Synopsys' gold-standard StarRC™ engine.

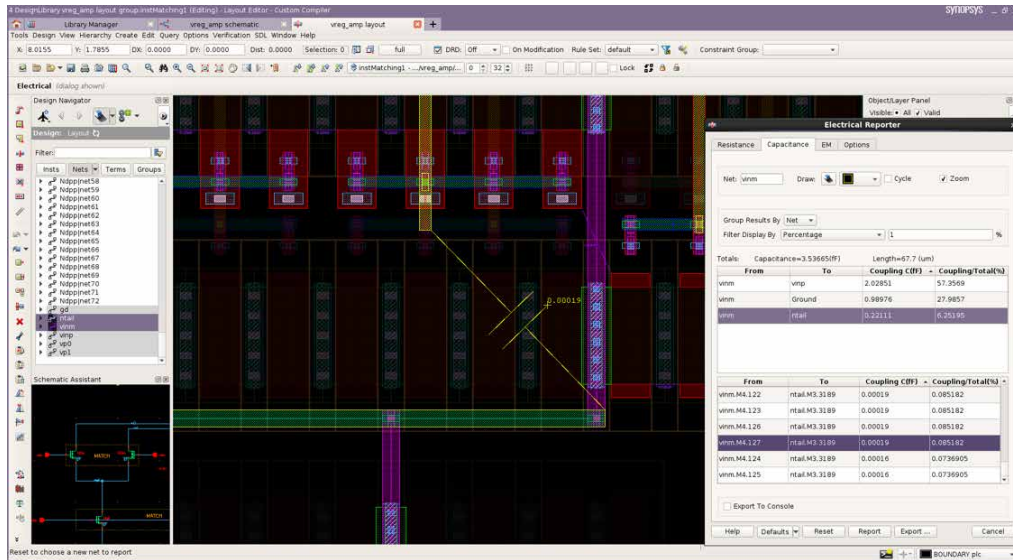


Figure 6: Electrical reporter checks resistance, capacitance, and electromigration during layout

Custom Compiler works seamlessly with IC Compiler™ II to provide a unified solution for custom and digital implementation. It can open, edit and save IC Compiler II databases at any stage in the design process. Users can freely move back and forth between Custom Compiler and IC Compiler II, using the commands of each to successively refine their designs. IC Compiler II users can perform full-custom edits to their digital designs with Custom Compiler. Likewise, Custom Compiler users can use IC Compiler II to implement custom digital blocks in their designs. The bi-direction flow between the two tools ensures that the digital and analog portions of the design remain synchronized.

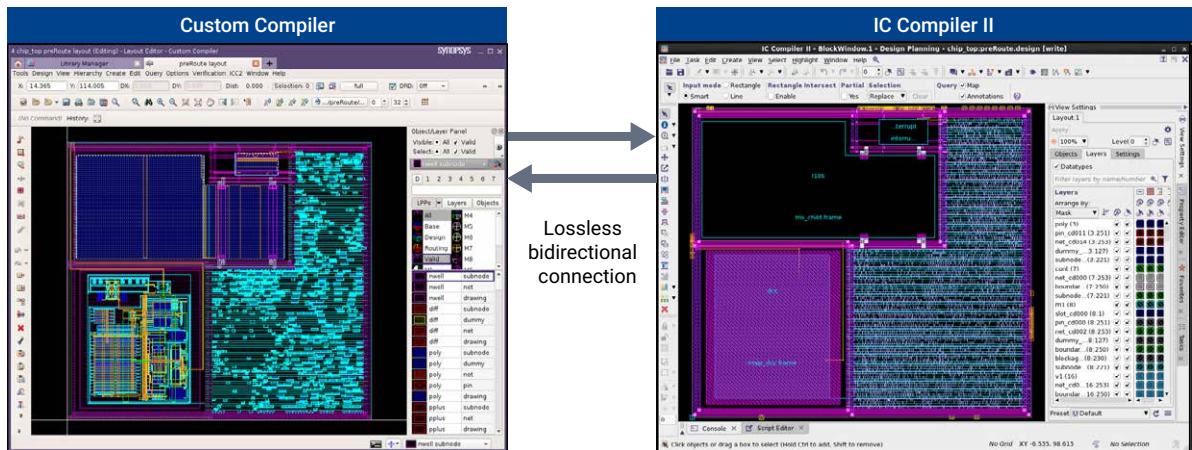


Figure 7: Custom Compiler can open, edit, and save IC Compiler II design databases

Design/Layout Collaboration

Custom Compiler includes features that make it easy to communicate design intent and achieve analog design closure.

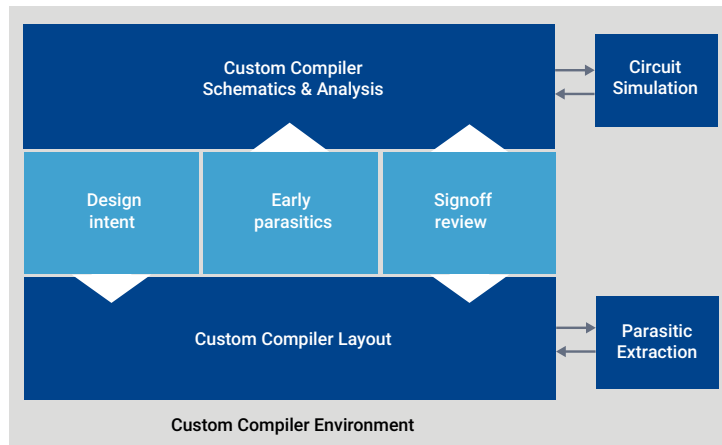


Figure 8: Custom Compiler supports close design/layout collaboration

For communicating design intent, designers can specify constraints on the schematic in the traditional way, or they can use templates, which are a powerful new way to specify groups of constraints. Designers can extract templates from previous designs for reuse. Templates include constraints for placement and routing patterns, dummy devices, and even guard rings. Designers simply choose the template they want to use in the layout—they don't need to enter constraints manually. Custom Compiler annotates currents and voltages from simulation onto the layout to enable electromigration and voltage-dependent design rule checking during layout creation.

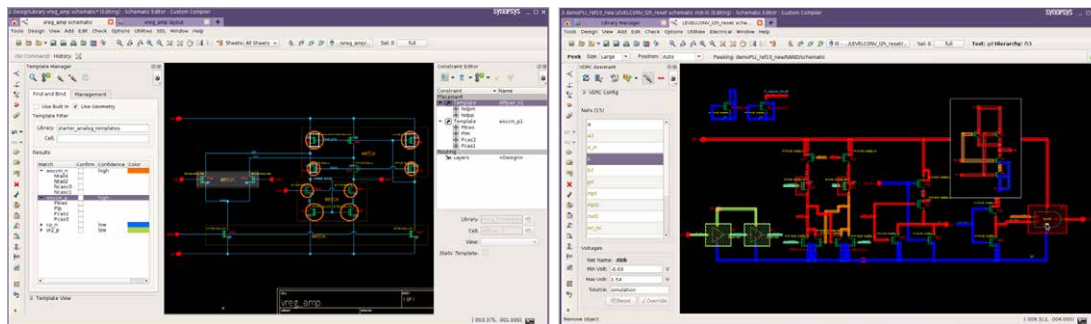


Figure 9: Design intent communicated via constraints (L) and current/voltage annotations (R)

The Custom Compiler Extraction Fusion and DRC Fusion technologies reduce the time it takes to achieve analog design closure. Extraction Fusion enables layout parasitics to be extracted from a partially completed layout. This provides circuit designers and layout designers with earlier feedback on layout parasitics. Circuit designers can use early parasitics to refine their designs and avoid layout rework. Layout designers can use early parasitics to confirm they are meeting design specifications. DRC Fusion enables live design rule checking during layout using IC Validator™. By checking for errors during layout, designers can reduce the number of late-cycle iterations caused by design rule violations discovered during final signoff checking.

Open and Customizable Environment

Custom Compiler is natively based on the OpenAccess database, so it is straightforward for users of non-Synopsys custom design tools to adopt. It includes a rich set of open APIs that make it easy to customize the user experience and integrate third-party tools. Third-party tool integrations are available for design data management, circuit simulation, physical verification, parasitic extraction, and other applications. Supported extension languages include TCL, Python and C++. Custom Compiler supports the interoperable process design kit (iPDK) standard. Custom Compiler iPDKs are available for most advanced node processes.

For more information about Custom Compiler, visit <https://customcompiler.info/>.