Overview
Helix is an advanced device-level automated placement solution for analog, and mixed-signal IC designs such as PLLs, SerDes, ADCs and PHYs. With Helix, users can start implementation earlier than with a traditional manual layout flow and save weeks of layout time, concurrently producing layouts of various aspect ratios in minutes so the user can decide the best floorplan and device placement for their circuit.

Introduction
Helix is a device-level automated placement solution for analog, and mixed-signal IC designs such as PLLs, SerDes, ADCs and PHYs. A designer working with Helix can produce very high-quality, design-rule-correct layout in a small fraction of the time needed using conventional methods, especially at 28nm and below. Helix is fully hierarchical and offers capacity in the tens of thousands of transistors, enabling it to optimize floorplanning and placement for entire custom IC blocks or IP at once (see Figure 1). Its output is a standard OpenAccess database that can be read and edited in any OpenAccess layout editor, such as Synopsys Galaxy Custom Designer®, Laker™, or Cadence Virtuoso®.

Figure 1: Helix dramatically shortens and improves the analog/mixed-signal layout cycle
Key Benefits

- Perform fast, accurate device-level floorplanning and high-quality placement
- Reduce analog/mixed-signal layout time and get early estimates of layout parasitics
- Port analog IP across technologies
- Automatically optimizes to complex design rules at 28nm and below, including support for:
  - Layout-dependent effects (LDE)
  - Dummy poly sharing
  - Density requirements
- Easy to use by circuit and layout designers
- Enhanced GUI accelerates many common layout tasks:
  - Constraint entry and editing using the schematic view. Users can drag devices from the schematic view to the constraint view, or right-click on devices in the schematic to create a new constraint
  - Cross-probing between schematic, layout and constraint views. Users can click on a block in the schematic to see the corresponding block in multiple layout views, or click on a constraint to see the included devices in the schematic and layout
- Dockable windows enable users to customize their environment
- Supports both iPDKs and Cadence SKILL PDks

Automate Custom Floorplanning and Layout at 28nm and Below

Starting with just a netlist, Helix can automatically generate the initial constraint set for a circuit (see Figure 2) and start creating placements. Within hours, Helix can produce an initial layout suitable for estimating parasitics and area, even if the circuit design is only partially complete.

Using this initial layout, designers can floorplan their design top-down, constraining size, pin locations and block relationships. Or they can start at the device level, building the layout bottom-up. In either case, the automated initial layout dramatically compresses the time needed to get the initial layout results.

Figure 2: Helix created initial PLL layout before circuit design was complete for the clock generator block

Helix is fully hierarchical and offers capacity in the tens of thousands of transistors, enabling it to optimize floorplanning and placement for an entire custom IC block or IP.

Accelerate Common Layout Tasks

Using Helix’s integrated environment, users can view constraints simultaneously with the schematic and the layout. Constraints may be entered and edited using the schematic view, or users can drag devices from the schematic view to the constraint view. In addition, users can right-click on devices in the schematic view to create a new constraint.

All views can be fully cross-probed, including schematic, layout and constraint views. Users can click on a block in the schematic to see the corresponding block in multiple layout views, or click on a constraint to see the included devices in the schematic as well as the layout.

GUI windows are dockable, enabling users to customize the Helix environment (see Figure 3).

Explore Layout Options and Refine Circuit Design

When layout is done manually, designers seldom get to see more than one floorplan of their circuit because of time constraints. Using Helix in a custom layout flow, designers are able to explore multiple layouts, make tradeoffs, and reduce unnecessary margins (see Figure 4).

Helix creates the full range of minimum-spacing layout alternatives that meet constraints and are DRC-correct. By creating these layouts quickly, Helix enables design teams to iterate the circuit and layout together, maximizing performance and area utilization.

With a capacity of tens of thousands of devices, Helix can handle entire...
analog and mixed signal function blocks without requiring designers to create dozens of small layout partitions. Using Helix to quickly explore multiple footprints, designers can typically save 5%-15% or more die size, compared with manual layout on a tight schedule.

Programmable Constraint Generation: XGen

The interaction between the constraint system and the layout engine is critical to the effectiveness and usability of any automation system. XGen is a general-purpose, programmable analog constraint generation and translation engine, now embedded within Helix, which can be scripted to automatically create a wide variety of Helix constraints. XGen is circuit-aware and has multiple applications, including:

- Automating constraint generation from netlists
- Allows users to capture their own design styles and higher-order "custom constraints" using XGen's programmable constraint templates

Automating Initial Layout with Helix First Look (HFL)

Built on XGen, HFL is an application that automatically generates the initial constraint set for a circuit with no additional input from the user. HFL uses process- and design-specific data from a configuration file to analyze the netlist and generate constraints. HFL enables designers to set default rules for automatic implementation.

The combination of accurate DRC handling and even a small number of general rules produces surprisingly good initial layouts. And unlike "gen from source" commands in other tools, HFL is fully hierarchical, handling designs with tens of thousands of devices. The placement it drives Helix to create is based on netlist connectivity, process design rules and predefined constraint sets, instead of symbol placement on the schematic.

HFL does not create final layout. Instead, it very rapidly creates initial constraints, dramatically shortening the time to get to initial layout, and eliminating much of the time-consuming drudgery of repetitive layout tasks.

Accelerate Existing OpenAccess Layout Flow

Helix fits easily into OpenAccess-compatible design flows. Inputs are a SPICE netlist and PDK. Helix can also read an OpenAccess schematic to enable schematic-based constraint creation and cross probing between schematic, constraint and layout views.

Helix output is a standard OpenAccess database that can be edited in any OpenAccess layout editor, such as Cadence Virtuoso 6.x, Synopsys Galaxy Custom Designer or Laker. Helix works with both PyCells and Cadence SKILL® PCells, in any combination. All Helix layouts are design-rule-correct.
Enabling Process-Portable Analog IP

By separating Design IP (netlist and constraints) from process data, Helix helps users to create process-portable analog IP (see Figure 5). Helix technology handles design rule requirements automatically. As long as you use a process-portable PCell library and your circuit topology remains the same, your design IP can be reused to quickly create a new layout on different processes or variations.

Platform Support

- Red Hat Enterprise Linux version 4 and 5 (AS, ES, WS)

For more information about Synopsys products, support services or training, visit us on the web at: www.synopsys.com, contact your local sales representative or call 650.584.5000.