









OrbitIO Planner

SYSTEM IO PLANNING AND OPTIMIZATION

Use OrbitIO Planner to:

-  Reduce iterations through coordinated chip-package-board design planning
-  Eliminate cost and complexity due to poor IO planning
-  Coordinate pin assignments for high speed interfaces like PCI-X, DDR, SerDes, etc...
-  Optimize placement and IO assignments for SiP, stacked die and PoP packaging
-  Evaluate wirebond feasibility during pad ring layout when changes are easily made
-  Optimize flip-chip designs to simplify RDL and bump escape routing
-  Quickly respond to feasibility requests using best available data
-  Optimize pad ring layout for application specific packaging

OrbitIO Planner™ revolutionizes the system IO planning process by unifying chip, package and board data in a single environment where placement and connectivity scenarios are easily derived and evaluated in the context of the full system. A unified chip-package-board data model immediately propagates changes to adjacent domains providing instantaneous feedback on their system-wide impact. With OrbitIO Planner, designers balance connectivity and IO assignments to optimize performance, cost and manufacturability prior to detailed implementation resulting in fewer iterations and shorter cycle-times.

Big Picture Design Planning

OrbitIO Planner can be used from early feasibility through detailed implementation. The ability to use best available data and create devices on-the-fly enables early stage design planning. Chip considerations can guide downstream connectivity, designs can be optimized for multiple packages, or the PCB socket can drive upstream IO decisions. As IO plans are refined, routability and manufacturability can be assessed. OrbitIO Planner replaces highly iterative and error-prone practices that have broken down with today's IO complexity.

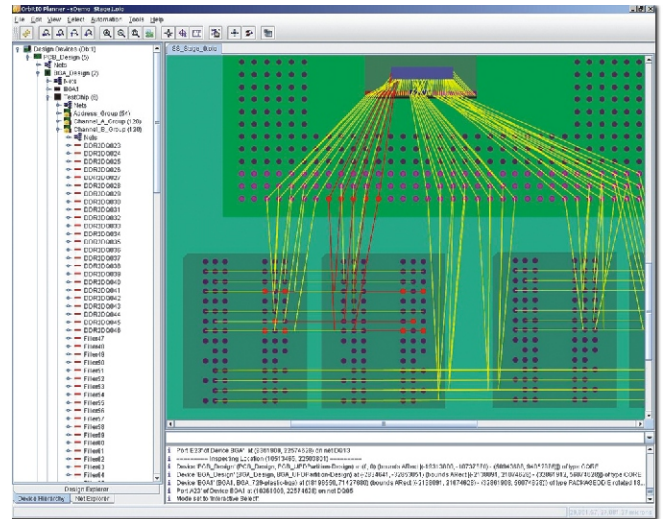


FIGURE 1. OrbitIO Planner optimizing chip and package IO assignments driven by the DDR2 memory on the printed circuit board.

End-to-End Connectivity

OrbitIO Planner delivers dynamic access to connectivity and placement data from gate to board level in a single tool. IO pad ring development is easily performed while simultaneously considering core and ball pad assignments guided by substrate specific rules with device and net-specific personalities. These personalities can influence pin assignment eligibility as well as proximity to on-chip power rails and clock domains.

OrbitIO Planner

SYSTEM IO PLANNING AND OPTIMIZATION

Vertically Aware Planning

System-in-Package (SiP), stacked-die and Package-on-Package (PoP) are increasingly popular integration strategies. OrbitIO Planner's vertically aware environment can be used to fully assess options for stack order, component rotation, and bonding of these multi-die designs. Users can optimize chip to chip connectivity or use the fixed IO of one chip to influence assignments on adjacent devices.

Multi-Domain Correlation

One of the challenges in IO planning is dealing with syntactical differences in net names between domains. It is common for a logical net to be referred to by three different substrate-specific names. OrbitIO Planner's powerful net management feature detects and maps such differences without the requirement for net list edits. OrbitIO Planner also includes a powerful correlation engine to map differences between logical and physical hierarchy in a way that avoids the need for Verilog or VHDL changes. OrbitIO Planner coordinates design refinements across the chip, package and board to quickly determine the impact of prospective changes on neighboring domains.

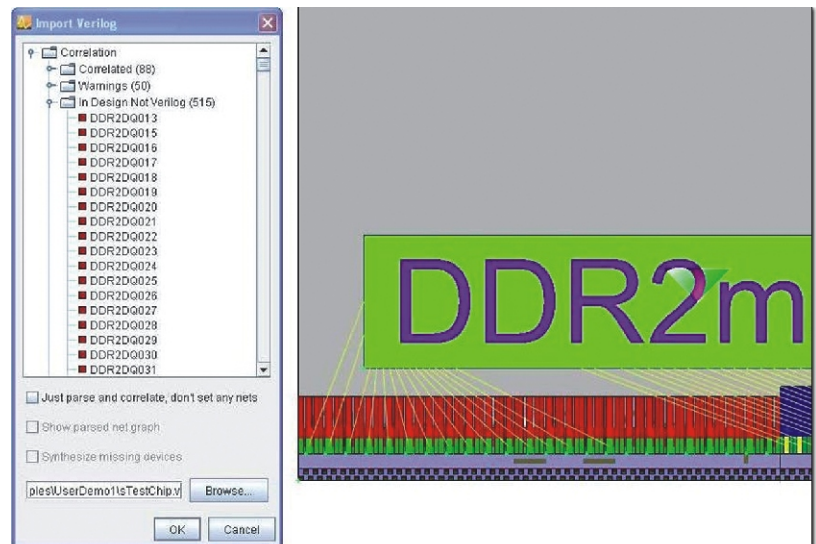


FIGURE 2: Verilog correlation between logical and physical data in OrbitIO Planner.

Interfaces

- Available for use with Windows and Linux.
- Chip Data: Bi-Directional LEF/DEF, Verilog, VHDL and CSV files.
- Package & SiP Data: Sigrity's UPD, Cadence, and Zuken.
- PCB Data: From most popular systems via DSN or .SPD.
- Bi-directional net list support.