interra systems

Concorde

High Performance RTL Compiler

Highlights

- High performance RTL technology – both behavioral as well as structural
- Efficient Processing for structural designs into Gate
- Well-defined, complete set of API functions for easy integration
- Generates macro/gate level netlist optimized for area
- Sufficient preservation of RTL information into Netlist - either directly or thought easy-to-use utilities
- Easy to customize error messages
- Support beyond synthesizable subset
- Capability of handling large designs
- Support for Technology Mapping
- RAM extraction capability from RTL even for arrays/records of more than two dimensions, keeping the inferred RAM as nearest to the RTL specification of the arrays/records
- Fully re-entrant flow supported

Key Advantages

- Fullest coverage of Industry Standard Synthesizable subsets in System Verilog, Verilog, and VHDL
- Fullest coverage in Mixed Language Synthesis
- Extensively field tested
- Backed by Interra's field-proven expertise in developing HDL analyzers and netlist object models

Concorde, Interra's high performance RTL compiler, combines state-of-the-art algorithms and data structures to bring you the fastest RTL compilation tool available today.

In Concorde, EDA tool developers find an efficient synthesis tool, which can be used to develop a wide range of EDA applications. Concorde offers analysis of design files, synthesis of RTL design description to a gate-level netlist, and a means to optimize logic, handle user-defined technology libraries, and manipulate the gate-level netlist to desired implementation.

What is New

→ Support for System Verilog IEEE 1800-2012 standard

→ Support for Slim Object Model

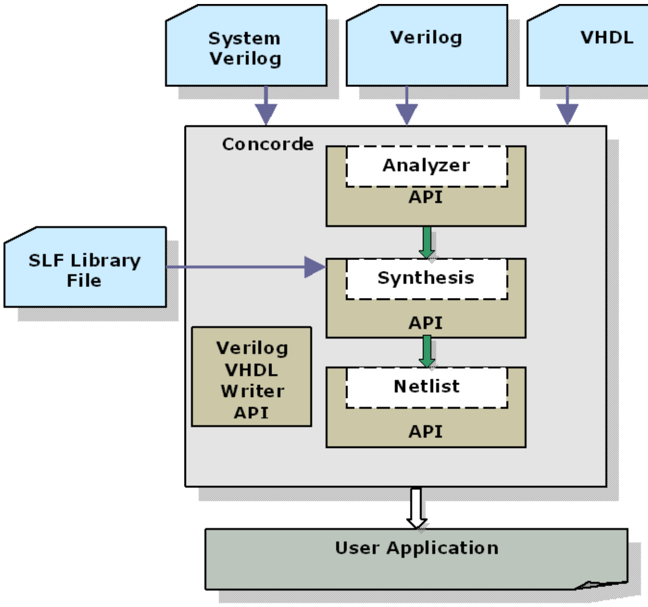
→ Improved run-time and memory performance

→ Support for enhanced messaging during elaboration and synthesis stages for easier debug

Concorde supports System Verilog, Verilog, VHDL and all combinations of the above languages as Mixed Language Synthesis Capability. Concorde has various switches to control the synthesis process. Concorde features include preservation of design hierarchy and names, netlist at complete gate level or at macro level, state re-encoding, name map information for debug usage, and preservation of X's and Z's in the design. You can control and customize these features for use in a wide range of application domains.

Concorde's architecture has built-in hooks for easy integration with user applications. Concorde can be used as a stand-alone binary with command line options, or there is a set of API functions, which can be used to perform synthesis through the user application.

Concorde is available on Solaris, Linux, and Windows platforms.



The Concorde Features

Supports Mixed SV-VHDL Designs

Concorde supports mixed System Verilog and VHDL designs with all the complex data types in ports and parameters.

Generates Macro/Gate Level Netlist

Concorde generates generic devices, such as adders, subtractors, multipliers, incrementors, decrementors, left/right shifters, comparators, decoders and muxes at the macro level. You can map these macros to desired implementation or map them to generic implementation, which are provided as default.

Provides Intuitive API Access

Concorde's API enables seamless integration of your application to fast RTL synthesis. Using the API, you can setup options, synthesize, and generate netlist. You can access, modify, and transform in - memory netlist for tighter integration. You can even access and modify the design before synthesis and elaboration.

Preserves Design Hierarchy and Names

Concorde preserves the hierarchies present in the design as well as the signal names ensuring ease of understanding and debugging of the generated netlist.

Keeps Name Map Information for Debug

Concorde provides slim and improved attribute mechanism, such as name mapping information, which is extended to optimized netlist. Concorde keeps the name map information from RTL to generated gates. Your applications can map gates/macros in the netlist to the corresponding RTL represen tation, allowing easy debugging .

Allows Technology Mapping

Concorde provides the support for complex user defined primitives during technology mapping. You can map the synthesized netlist using user-defined technology library cells for customized implementation of generic gates. This feature allows you to generate efficient and area-optimized implementation of your designs. You can even translate designs from one technology to another. Your applications can also use user-defined technology library to map synthesized netlist for customized implementation of generic gates.

Preserves X's and Z's in the Designs

Concorde provides you the option to preserve explicit assignments to X's and Z's in the RTL design.

Efficient Transformation of Structural Designs into Gate Level

Concorde has the capability of transferring structural designs efficiently into Gates. However the interface would remain the same. So you do not have to bother about the 'type' of the design, - just call Concorde synthesis interface. It would do the rest!

Reads Library Files in SLF format

Concorde can read library files in SLF format and can map these library cells to primitive gates. These library cells can be used for linking and mapping to the synthesized netlist.

Allows Memory Extraction

Concorde extracts RAM from an RTL design to a separate, black box module. You can decompile the RTL body of the memory along with structural netlist. This mechanism helps in simulated verification of generated netlist. At the same time, this mechanism can potentially reduce Concord's memory footprint.

You can accept/discard the synthesized RAM. If you discard the synthesized netlist, you can just call the same synthesis Interface of Concorde to re-synthesize it in standard logic cells.

Handles Large Designs

Concorde can synthesize large design descriptions. Enriched set of options allows incremental analysis, disk dump, and synthesis. Additionally, Concorde handles large designs more efficiently than before with large arrays used across number of processes/always.

Allows Customization of Error Messages

Customized message handling allows you to tailor information, warning, and error messages to the needs of your application. You can suppress messages and even register user-defined message handlers. With this feature your application can have a consistent interface.

Interra Systems, Inc.
601 S. De Anza Boulevard, Suite 212, Cupertino, CA 95014
Phone: +1 408 579 2000   Fax: +1 408 579 2050
Email: info@interradesign.com   Web: www.interradesign.com