

Concorde

High Performance RTL Compiler



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.

Concorde supports System Verilog, Verilog, and VHDL; the industry standard synthesizable subset; and synthesis options. Concorde also supports mixed language design flow. 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.

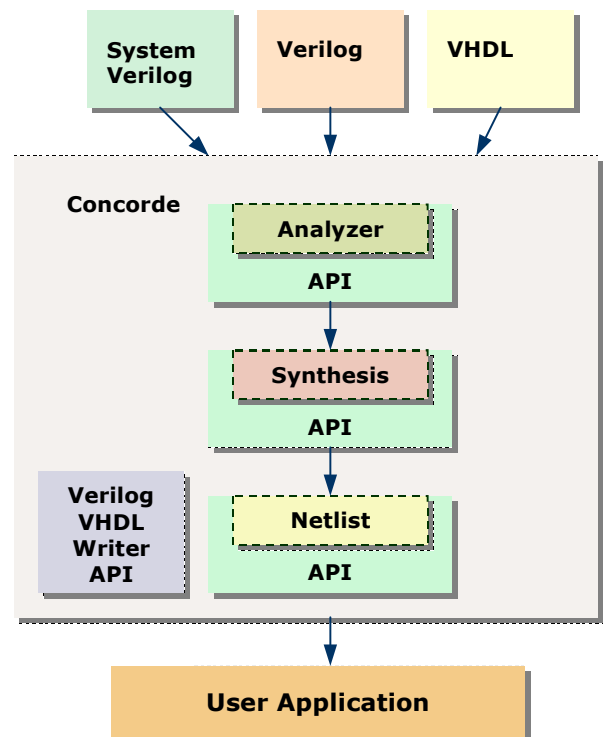
Concorde is available on Solaris, Linux, and Windows platforms.

Key Advantages

- Full support for System Verilog, Verilog, and VHDL
- Additional support for mixed design flow
- Complete support for industry standard synthesizable subset
- Extensively field tested
- Backed by Interra's field-proven expertise in developing HDL analyzers and netlist object models

Highlights

- High performance RTL technology
- Well-defined, complete set of API functions for easy integration
- Generates macro/gate level netlist optimized for area
- Preserves design hierarchy and names
- Name map information for debug usage
- Easy to customize error messages
- Compatible latch and tristate inferencing
- Support beyond synthesizable subset
- APIs to manage memory usage
- Module-by-Module and Hierarchical Compilation flows
- Capability of handling large designs
- Support for Technology Mapping



Concorde Features

Supports System Verilog

Concorde supports the synthesizable subset of System Verilog constructs including multi dimensional Array, interfaces, structure/unions, uniquepriority, and enhanced loop.

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 keeps the name map information from RTL to generated gates. Your applications can map gates/macros in the netlist to the corresponding RTL representation, allowing easy debugging.

Allows Technology Mapping

Concorde allows you to 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.

Provides Module-by-Module and Hierarchical Compilation Flows

Concorde provides you the flexibility to control synthesis; synthesize specific modules/architectures, ignore modules, execute hierarchical synthesis, and more.

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 memory 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.

Handles Large Designs

Concorde can synthesize large design descriptions. Enriched set of options allows incremental analysis, disk dump, and synthesis. Additionally, Concorde can synthesize a structural design efficiently with optimum use of memory and time.

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.

Also available:

Jaguar, Cheetah: HDL Analyzers

NOM: Netlist Object Model

Tiger: Static Timing Analyzer