MixedHDL

The Mixed Language Digital Design Elaborator

What is New

→ Support for multithreading compliant APIs

Highlights

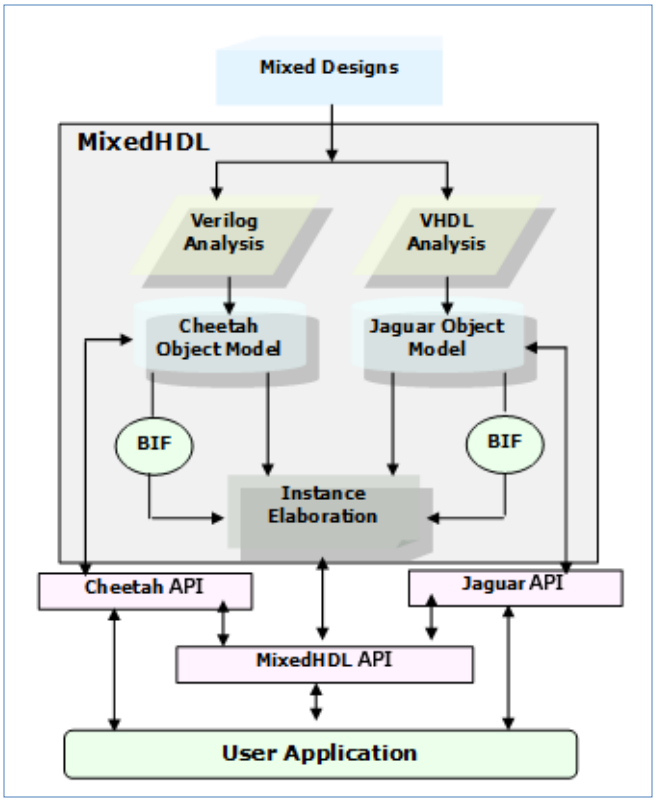
- Latest support of language standards through Cheetah and Jaguar
- Configurable Elaborator
  - Light weight single level step-by-step elaboration
  - Static elaboration of the whole design
- Static elaboration provides for
  - Uniquification
  - Unrolling
  - Multi-level scope variable resolution
  - Multi-level defparam propagation
  - Resolution of parameter/generic overrides
  - Semantic check for elaborated object model

Most of the complex SoC designs today use one or more of Verilog, System Verilog, and VHDL to implement various sub-systems. EDA tool developer needs to provide support for mixed language parsing.

MixedHDL from Interra Systems uses Cheetah, the Verilog and System Verilog front-end analyzer, and Jaguar, the VHDL front-end analyzer, to bring MixedHDL capability for EDA tool developers. Programming layer of MixedHDL analyzes mixed designs and elaborates cross-HDL instances. EDA tool developers can bring mixed-language support to their tool quickly. Language specific functionality is available through Cheetah and Jaguar API layer.

MixedHDL in conjunction with Cheetah and Jaguar parses all versions of standards of Verilog, SystemVerilog and VHDL and creates an in-memory Object Model. Applications can then access the design information instantly using efficient generic programming interface. Standard compliant and easy to use building blocks from Interra Systems enable users to reduce time-to-market by saving a significant amount of development time.

MixedHDL is available on Solaris, Linux, and Windows NT platforms.



API: Application Programming Interface

The MixedHDL Features

Complete Language Support

MixedHDL is built over Cheetah and Jaguar analyzers. Cheetah completely supports System Verilog IEEE 1800-2012 and IEEE 1800-2009. Cheetah is backward compatible with IEEE 1800-2005, IEEE 1364-1995, IEEE 1364-2005, IEEE 1364-2001, OVI 2.0, and simple subset of IEEE 1850, V1.1 and V1.01.

Jaguar supports IEEE-1076-2008. Jaguar is backward compatible with IEEE-1076-2002, IEEE-1076-1993 and IEEE-1076-1987.

Cheetah and Jaguar analyze the syntax and semantics of mixed designs. In addition, MixedHDL provides a facility to elaborate mixed digital designs covering different aspects of System Verilog and VHDL.

Complete Set of API Functions

MixedHDL has specific APIs to elaborate mixed designs. These APIs enable the user to find out master of an instance across the language boundary. MixedHDL APIs work on both Cheetah and Jaguar Object Model. Therefore, one can use MixedHDL APIs in addition to Cheetah or Jaguar APIs in their applications. While the Cheetah and Jaguar APIs let users work on System Verilog and VHDL designs, respectively, MixedHDL enables the user to elaborate mixed designs. Using MixedHDL APIs, one can work in two modes: the in-memory mode and the dump mode.

Light weight, single level, step by step Elaboration

MixedHDL APIs provide the facility for elaborating a master design unit only when required. Further, the APIs can be used to un-elaborate the master design unit and re-elaborate it with a different set of parameter values. With this feature, MixedHDL users can effectively keep memory requirement of the application low.

Static Elaboration of the whole design

MixedHDL enables a full static elaboration of mixed designs. A master design unit can be elaborated, copied, and instances can be bound to the copied master design unit across language boundaries. This flow traverses the Design Hiererarchy in DFS mode and flattens a design that has generate blocks and array of instances, propagates defparam values and resolves scope variables across language boundaries. MixedHDL also supports SV Bind construct elaboration where the target scope is both SV and VHDL in full elaboration flow.

Black Box Elaboration

MixedHDL allows elaboration of an instance whose master is not defined. MixedHDL creates a dummy design unit (module if the instance language is System Verilog or entity if the instance language is VHDL) with a list of input and output ports inferred through corresponding instantiations.

Customization of Errors

The API functions enable the user to customize the messages to suit application-specific needs. One can also register an error message handler to meet customized needs. In addition, one can use the API functions to suppress error messages and warning messages or change the severity of messages.

Interra Systems, Inc.
1601 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