

Highlights

- Latest support of Language standards
 - VHDL: IEEE-1076-2008, IEEE-1076-2002, IEEE-1076-1993, IEEE-1076-1987
 - PSL: IEEE-1850-2005, V1.1 and V1.01
- Comprehensive syntax and semantic checks
- Language compliant library management for incremental compilation
- New Generic Programming Interface for efficient EDA application development
- Browser and Decompilation utility for easy debugging
- Editable and Extensible object model
- RTL subset semantics checks
- Utility Objects for Expression Evaluation, Elaboration and Partial analysis
- Capability to scan and sort based on analysis order
- Support for:
 - Static Elaboration
 - Complete source information including file name, line/column number for each construct, and user comments
- Highly customizable - error handler, user-defined attributes, user-defined meta comments and more

Addressing the needs of EDA tool developers who need to accelerate deployment of VHDL products, Interra offers Jaguar—a high performance VHDL analyzer. Jaguar is targeted as a customizable front-end for VHDL based applications, such as simulators, schematic generators, synthesizers, and code generators.

Jaguar is field proven and offers a best value solution. Jaguar analyzes the VHDL design description in one pass and generates a binary intermediate form, which is accessible through a well-defined intuitive Generic Programming Interface (GPI). The GPI provides an easy-to-integrate VHDL front-end to EDA applications. EDA tool developers can use the GPI to analyze as well as access designs for information, modify designs, evaluate expressions, perform elaboration, and optionally check for RTL subset compatibility.

Jaguar also supports PSL analysis for both embedded PSL and external verification units. Jaguar offers control options enabling applications to work with PSL and VHDL together.

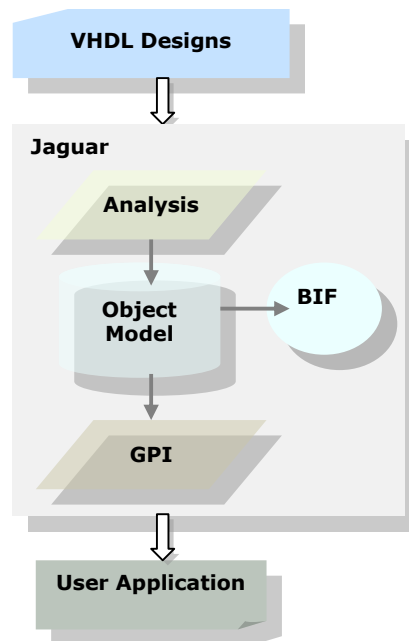
Jaguar is available on Solaris, Linux, and Windows platforms.

What is New

- C++ based Generic Programming Interface
- Supports for VHDL 2008 (IEEE 1076-2008)

Key Advantages

- High performance VHDL analysis
- Option to comply with any of supported standards
- Compile time type-safety helps catch errors early
- Usage of high level design patterns makes application code more concise
- Separation of flow and actions to make EDA tool development more maintainable
- Backed by Interra's field-proven expertise in developing HDL analyzers



GPI: Generic Programming Interface

The Jaguar Features

Complete Language Support

Jaguar supports VHDL IEEE 1076-2008, including the features like integrated PSL in VHDL, external names, new logical, relational and conditional operators, if and case generate statements, matching case statements, new assignment statements, process all, expressions in port map and more.

Jaguar is also backward compatible to VHDL IEEE 1076-2002, IEEE 1076-1993 and 1076-1987 versions. Jaguar performs very strict semantic checks on all language constructs. You can even switch off some semantic checks during analysis to reduce run time.

Jaguar also supports analysis of simple subset of IEEE 1850, V1.1 and V1.01. After analysis, both PSL objects and VHDL objects co-exist in memory.

Concise Set of GPI Functions

Jaguar provides a concise set of Generic Programming Interface (GPI) functions which you can use to access or manipulate information on the object model. You can also dynamically extend and modify the object model and work with the in-memory objects.

Powerful Object Model

The Jaguar Object Model (OM) is derived from the VHDL language grammar so that accessing the information from the OM is an intuitive process. The OM keeps direct reference to an object in an external design unit, saving significant runtime during the elaboration phase. Also, the OM stores the sizes and values of expressions whose sizes are determined at compile time.

Support for RTL-based Applications

Jaguar provides a runtime switch to perform RTL subset checking on a given VHDL description. Jaguar performs subset checking in compliance with industry standard synthesis policy. All compliance errors are reported either as an error or a warning, depending on the severity. If the design is RTL compliant, Jaguar annotates the object model with synthesis-specific information including inferred clock, set and reset signals, and expression sizes.

Elaboration

You can use Jaguar GPI functions to elaborate a design unit. You can elaborate a fully instantiated design and evaluate the size and values of all globally static expressions, including function calls.

Debug Tools for Development

Using the GPI, you can easily decompile and browse any node in the object model.

The Decompiler utility enables you to write the complete in-memory object model as a VHDL description.

The Browser utility helps you in better understanding of object model traversal.

The Browser prints out object type and its property names and values. It also allows you to navigate Jaguar object model easily from a given System Verilog example and do partial decompilation of object from any given point of the syntax tree.

You can use the Browser to learn quickly the application programming requirements using Jaguar.

User-specific Meta Comments

Jaguar stores user-defined meta-comments. You can not only access the meta-comments, but also specify user-defined meta-comments that are to be processed by Jaguar.

Customization of Errors

Jaguar performs complete syntax and semantic checks on the design and detects as many errors as possible in the design in one pass analysis. You can easily customize the error messages for application-specific needs.

Expression Evaluation

You can use Jaguar GPI functions to evaluate and access the value and sizes of all locally static and globally static expressions, including functions.

Encrypted Code Handling

Jaguar handles encrypted code region inside a VHDL file. Jaguar provides a handle to your application to decrypt the code in the encrypted region and then populates the object model corresponding to it.