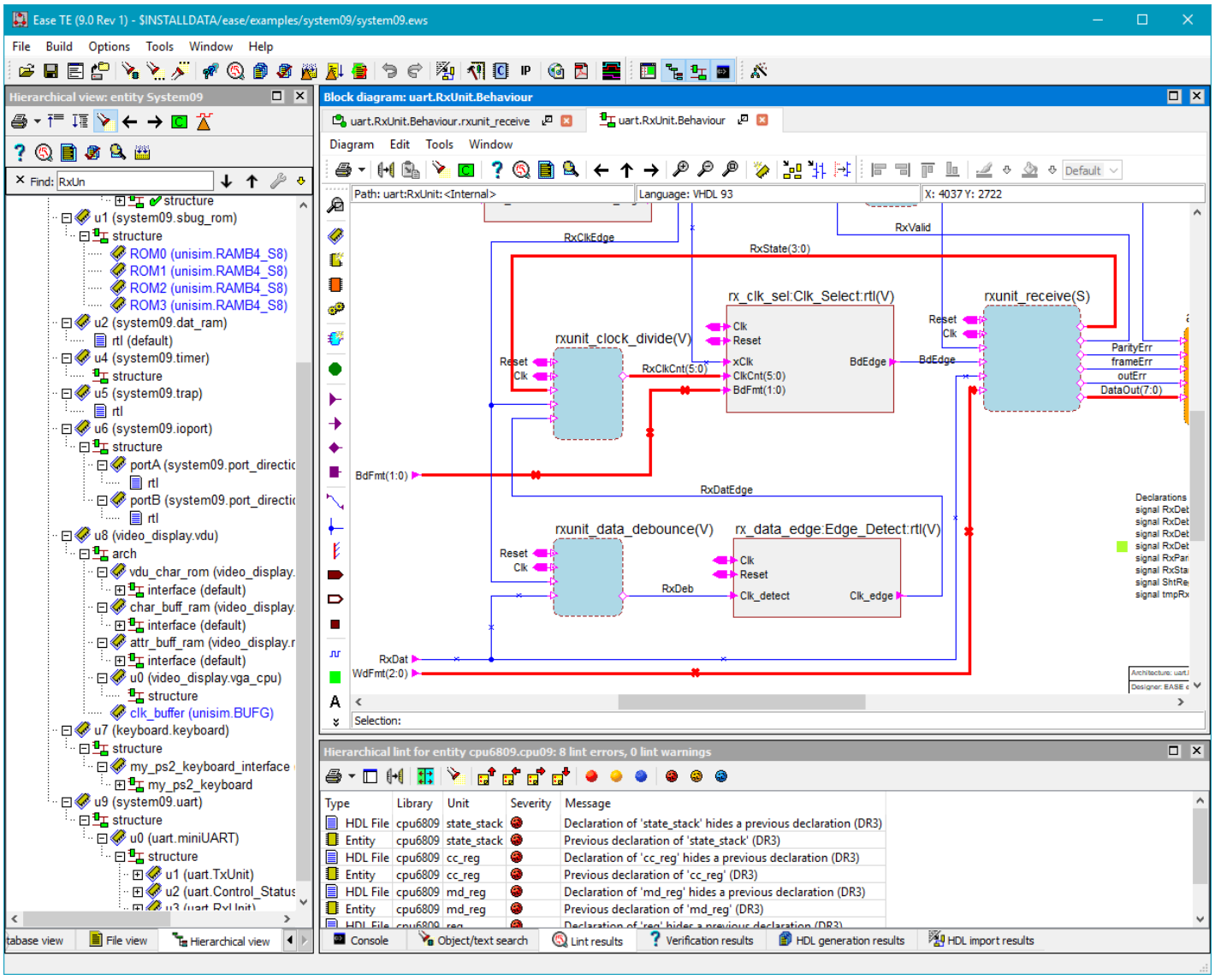
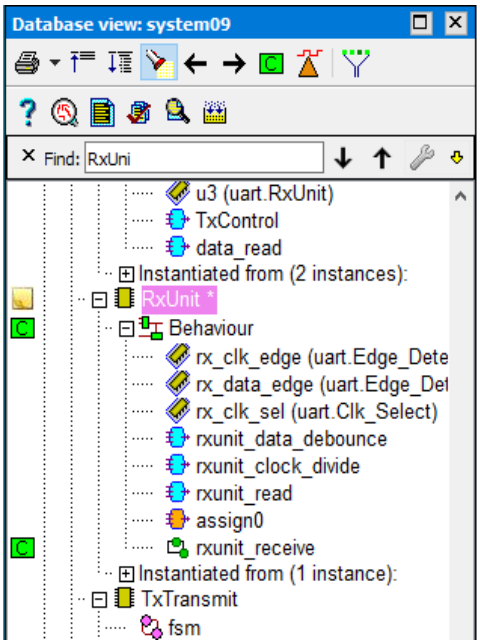
EASE Graphical HDL Entry

The ideal graphical environment for
developing, understanding and maintaining
complex HDL designs

As an experienced HDL designer you will know that an FPGA or ASIC design takes more than writing HDL code only. You will have to deal with the challenges of implementing changes that have impact on multiple places in the design hierarchy, exploring different implementations, documenting the design, integrating IP and/or functions from core generators, design reviews, working with a team of designers, etc. All these time consuming tasks are necessary, but they leave you with less time for the job you want to focus on: the actual HDL design work.
EASE is a design entry environment that enables you to create and manage HDL projects. It will help you to perform these time consuming tasks, so you can keep your mind on the design and implementation.



Project management

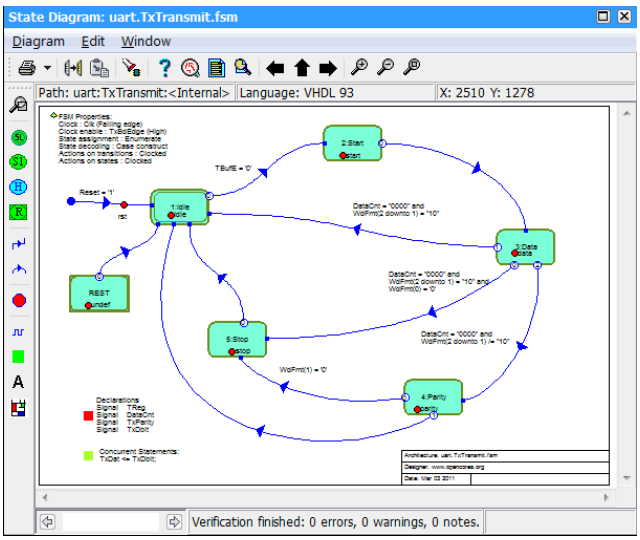


The project browser provides a good overview and offers easy access to the design elements.
The browser offers three views: the database view shows a tree of all elements in your project, the file view shows all generated files and the hierarchical view shows the HDL hierarchy of your project.
The browser also provides many details of the different objects, like verification status, “instantiated from” info, version number, attached documentation items and more. From the browser, all objects can be opened in their respective editor (block diagram, state diagram, truth table or text editor).

Editors

EASE features four editors, each targeted at a different implementation method.

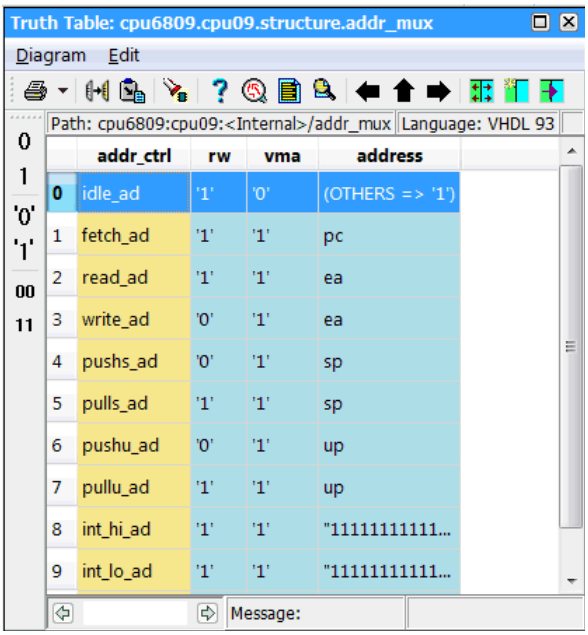
The block diagram editor allows you to easily decompose your system into functional blocks. EASE is built around VHDL and Verilog, allowing you to use the power of the languages by using constructs like user defined types, generics/parameters, block/ generate statements and configurations. The entities or modules can be edited directly in the block diagram or using a spreadsheet like editor. The internal database will ensure that each component is consistent with its entity or module. Changes to the IO can easily be propagated to any level in the hierarchy. The generate block (and diagram) allows you to instantiate components conditionally or as an array of components. Each block can be implemented using one of the four available editors. Facilitating an abstraction level between block diagrams and plain HDL code, the block diagram editor allows you to graphically represent VHDL processes or Verilog always statements. They can be implemented using state diagrams, truth tables or HDL text. This approach visualizes the data flow inside a single diagram.



The state diagram editor is used to describe control logic for which a finite state machine is the appropriated form. The editor supports Moore, Mealy and mixed state machines. Any valid VHDL expression or Verilog statement can be used to define transition conditions and actions. Transitions can be synchronous or asynchronous; outputs can be clocked or combinatorial. The editor supports a variety of state assignment methods, including binary, gray, one-hot and user defined. Within the state diagram editor you can also define Master Slave state machines. This is a special form of concurrent state machine where the master machine waits until a slave has finished. EASE will generate all the required synchronisation signals and logic.

The truth table editor is useful for decoders and decision logic. The spreadsheet like editor in combination with a flexible and smart use of column headers allows a compact

visualisation of the intended behaviour. A column-fill wizard is available to generate data in various encoding styles and representations.



The text editor was specifically designed for HDL language support and offers features like syntax highlighting, identifier and template expansion, block and column manipulation and in- and out-commenting. It is also possible to specify your own favourite HDL editor.

External IP

External HDL files like IP, legacy code, or FPGA generated models can be integrated in your project as external objects. EASE will create symbols and component declarations for each instantiated module. Symbols can easily be updated to reflect the latest version of your code. Existing HDL can also be translated into block diagrams. Symbol libraries for FPGA primitives can be created on the fly from vendor VHDL or Verilog descriptions.

Design flow

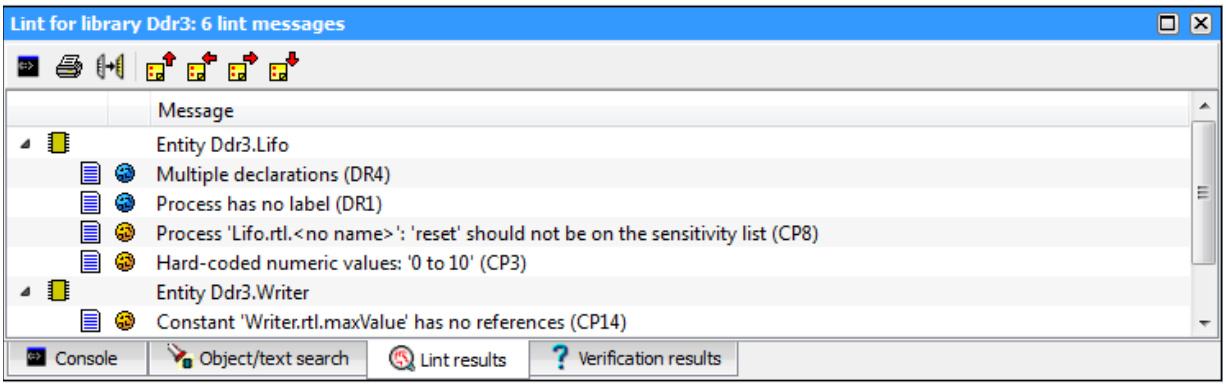
EASE offers easy integration with third party EDA software (like simulators, synthesizers and FPGA place & route tools). After specifying your tool flow additional tool buttons are added to the user interface for each tool you selected. With the buttons you can compile and simulate your design with the selected simulator or start your FPGA environment. The design flow can be tailored using the Tcl command language. A command line program to generate HDL code is also available to facilitate automated builds.

Documentation

EASE provides documentation of diagrams in various graphical formats like JPG, PNG, TIFF and PDF. Your complete project can also be documented in HTML format together with SVG diagrams. The HTML and SVG include hotlinks to browse through the design hierarchy and offers access to the generated HDL code in HTML format.

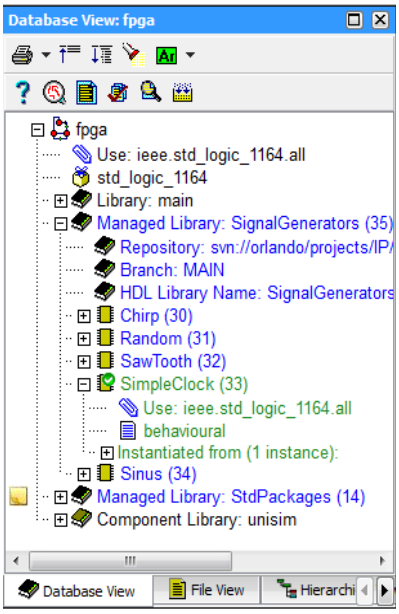
Verification and linting

Before VHDL or Verilog is generated EASE verifies the design for inconsistencies and syntax errors. Linting is an additional verification effort to find potential design problems (like range mismatches in assignments of vectors, or read-only signals) and improve



design scalability and quality. The linting is based on the DO-254 'Best Practice Design Rule Set'. The messages are hot-linked to the corresponding editor so you can quickly navigate to the offending code.

Team based design



Many FPGA's and ASIC's are designed by a team of engineers that need to work closely together to finish the implementation correct and on time. The best way to work together on a project is by using a design environment that allows a group of designers to simultaneously work on the project without interfering with each other. EASE supports team based design using the open source version management system Subversion. The Subversion client is integrated into EASE to achieve the best performance. A 'diff' command can be installed to see the differences between the generated HDL of two versions.

Creating IP

Using 'Managed Libraries' it is easy to build your own IP blocks and use them in various projects. The IP blocks can be checked-out in the local project (which will set a lock in the original project). If you commit any changes they will become available for anyone using this branch. A managed library can reside in another Subversion repository and you can select which branch or tag to use in your project. Multiple versions of the same managed library can be used in a project.

Productivity features

- Propagate name and type changes through the design hierarchy
- Add and delete ports and wires through the design hierarchy
- Automatically keep all instances of entities/modules up to date
- Edit entity/module symbols in the block diagram editor
- Generation of testbench skeletons
- Trace signals through the design hierarchy
- Push down re-factoring
- Master / Slave state machine synchronisation

Operating Systems	System Requirements	License Configuration
• Windows (64 bit) 7 / 8.1 / 10 • Linux (x86 PC, any recent 64 bit distribution)	• 150 MB free disk space • 2 GB system RAM recommended	• Nodelocked • Floating • FlexLM protected

HDL Works

HDL Works B.V.
Maxwellstraat 17
6716 BX Ede
The Netherlands
Phone: +31 (0)318 642 022
E-mail: info@hdlworks.com
http://www.hdlworks.com