

EDA Simulator Link 3.3

Verify VHDL and Verilog using HDL simulators and FPGAs

Introduction

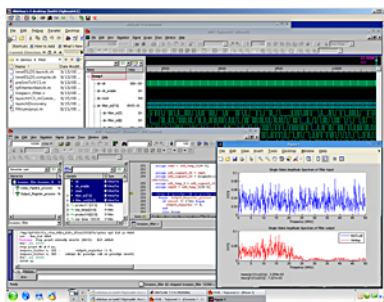
EDA Simulator Link™ provides a verification interface between MATLAB® or Simulink® and your HDL simulator or FPGA board. Using EDA Simulator Link you can verify a Verilog® or VHDL® design against your Simulink® model or MATLAB algorithm using cosimulation with a Verilog or VHDL simulator, such as Mentor Graphics® ModelSim® or Cadence® Incisive®. EDA Simulator Link also lets you perform hardware verification on your FPGA board using FPGA-in-the-loop simulation.

With EDA Simulator Link you can use MATLAB code and Simulink models as a test bench that generates stimulus for an HDL design and analyzes the simulation's response. You can replace HDL design components with MATLAB code and Simulink models, enabling simulation of the complete system before all the HDL design elements are available.

EDA Simulator Link lets you create transaction-level model (TLM) components for use in virtual prototyping environments.

Key Features

- Full VHDL, Verilog, and mixed-language cosimulation support for MATLAB or Simulink
- Test bench capability, enabling the use of MATLAB code or Simulink models to stimulate HDL code and check its response
- Component capability, enabling simulation of MATLAB code or Simulink models in place of entities not yet coded in HDL
- Cross-platform cosimulation using MATLAB or Simulink on one platform and the HDL simulator on a different platform
- Interactive or batch-mode cosimulation, debugging, testing, and verification of HDL code
- Single-machine, multiple-machine, and cross-network cosimulation using shared-memory or TCP/IP-socket communication modes

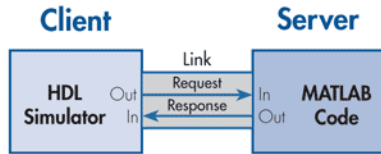


Elaboration of a floating-point reference algorithm and verification of a Verilog implementation using a cosimulation interface.

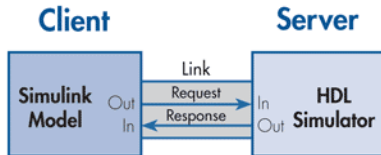
Working with EDA Simulator Link

EDA Simulator Link supports both MATLAB and Simulink environments, various user setups, and cosimulation scenarios.

MATLAB and Simulink communicate with HDL simulators via a bidirectional cosimulation interface with a client/server architecture.

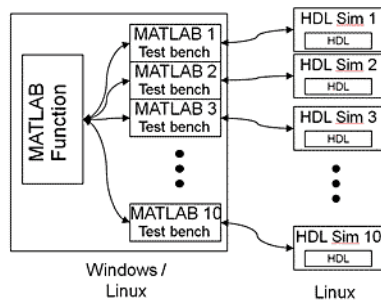


Bidirectional communication between the HDL simulator (client) and MATLAB (server).



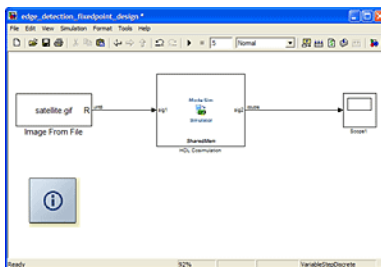
Bidirectional communication between Simulink (client) and the HDL simulator (server).

EDA Simulator Link enables interactive and batch-mode cosimulation on a single computer, across heterogeneous platforms, or across a network.

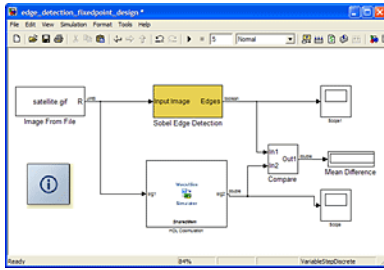


Interfacing multiple MATLAB test benches or Simulink cosimulation blocks to one or more HDL simulations. This approach supports one-to-one, one-to-many, many-to-one, and many-to-many configurations; it also supports multiple operating systems and computing platforms.

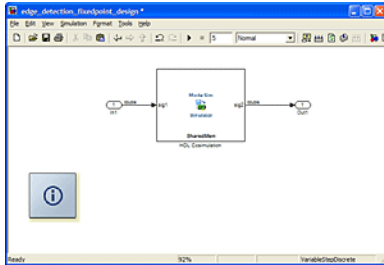
EDA Simulator Link enables three cosimulation scenarios using supported HDL simulators. Two of these scenarios involve a design under test (DUT), which can replace or be compared with a golden reference for the algorithm in the cosimulation. This golden reference is either MATLAB code or a Simulink model. In the third scenario, standalone Simulink cosimulation blocks can be used to instantiate existing HDL implementations into a Simulink model.



Cosimulation scenario using EDA Simulator Link: replacing an algorithm golden reference with a DUT.



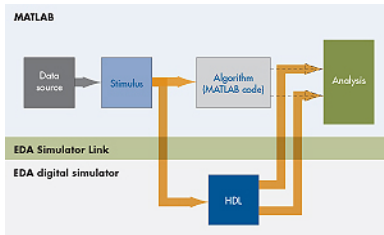
Cosimulation scenario using EDA Simulator Link: comparing a DUT with an algorithm golden reference.



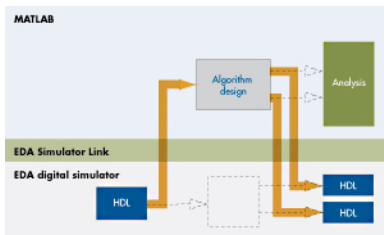
Cosimulation scenario using EDA Simulator Link: instantiating an HDL implementation into a Simulink model.

Typical Workflows

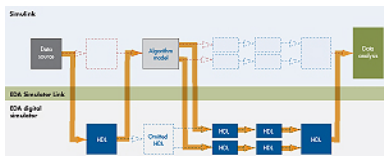
EDA Simulator Link supports three workflows: MATLAB test bench, MATLAB component, and Simulink cosimulation.



Using MATLAB code as a test bench supplying on-the-fly stimulus and analysis for an HDL simulation.



Substituting MATLAB code for an HDL component in an HDL simulation.



Mixing a Simulink system model with an HDL simulation as a bidirectional test bench and component.

Each workflow offers full interactive debugging capability in both client and server, enabling simultaneous use of the full features of the MATLAB or Simulink and HDL simulator environments, including viewers, breakpoints, and pattern matching. EDA Simulator Link can interface with multiple HDL entities and multiple HDL simulators from a single MATLAB test bench or Simulink model. Multiple MATLAB test bench functions and Simulink cosimulation blocks can also be used with one or more HDL simulator.

Using EDA Simulator Link, you can set up an efficient environment for cosimulation, component modeling, analysis, and visualization. In this environment, you can:

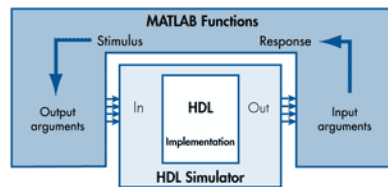
- Develop software test benches for HDL entities in MATLAB or Simulink
- Include HDL models in larger-scale system models developed and simulated in Simulink
- Perform interactive GUI, batch-based, or shell-based debugging
- Verify, analyze, and visualize HDL implementation simulation results in MATLAB and Simulink
- Generate functional test vectors for manufacturing test programs and verification

Using EDA Simulator Link with MATLAB

With the MATLAB component and MATLAB test bench capabilities, the HDL simulator is the client and MATLAB is the server. The time wheel resides in the HDL simulator because MATLAB algorithms are untimed.

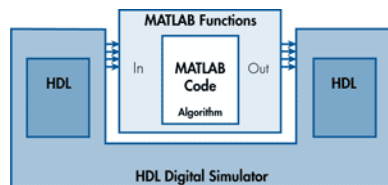
You can connect an HDL simulator to multiple MATLAB components or test benches running on one or more MATLAB servers. A MATLAB server can interface to multiple HDL entities and HDL simulators.

Using the MATLAB test bench capability, you can reuse the MATLAB test bench that was created to verify your MATLAB algorithm to verify your HDL implementation of the algorithm.



MATLAB code as a test bench for an HDL simulator. MATLAB code drives and analyzes the implementation while the block, an HDL implementation, runs in the HDL simulator.

Using the MATLAB component capability, a MATLAB algorithm takes the place of entities not yet coded in HDL, enabling simulation of the complete system before all the HDL design elements are available.



MATLAB code as a component for an HDL simulation. An algorithm coded as a MATLAB function takes the place of entities not yet coded in HDL.

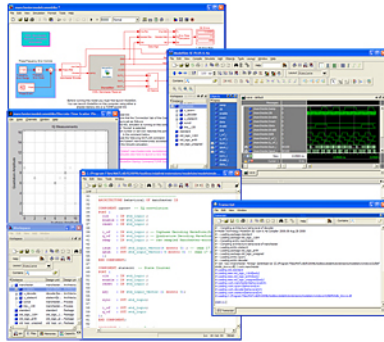
Using EDA Simulator Link with Simulink

The Simulink cosimulation capability lets you connect an HDL simulator to a Simulink cosimulation block. Simulink is synchronized with the HDL simulator by scaling Simulink time to HDL simulator time. You can connect a Simulink model with one or more cosimulation blocks to one or more HDL simulators.

Using Simulink and related blocksets, you can create system-level representations of signal processing, communications, and other systems. These models can be used to create software test benches for HDL implementations or substituted as algorithms in place of HDL blocks or subsystems for HDL debugging or verification.

EDA Simulator Link also supports verification tasks, including:

- Automatically scripting the HDL code compilation and simulator startup
- Using Autofill to create the cosimulation signal interface between the HDL code and Simulink model
- Maintaining System-C library compatibility through support of vendor-distributed GCC libraries



Cosimulation and software test benching of a Manchester Receiver, created using Simulink, DSP System Toolbox™, and Communications System Toolbox™. The Manchester Receiver is implemented in VHDL and simulated in ModelSim®.

Supported HDL Simulators and FPGA Boards

EDA Simulator Link supports HDL simulators and FPGA boards from multiple vendors. Information on [supported verification solutions](#) is available.

Resources

Product Details, Demos, and System Requirements

www.mathworks.com/products/eda-simulator/

Trial Software

www.mathworks.com/trialrequest

Sales

www.mathworks.com/contactsales

Technical Support

www.mathworks.com/support

Online User Community

www.mathworks.com/matlabcentral

Training Services

www.mathworks.com/training

Third-Party Products and Services

www.mathworks.com/connections

Worldwide Contacts

www.mathworks.com/contact