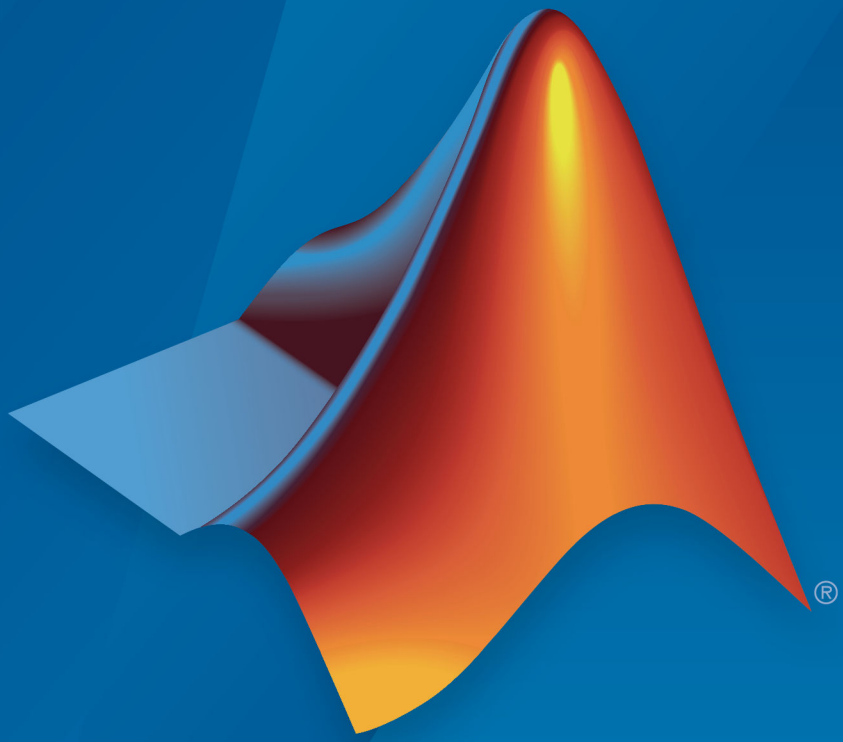
HDL Verifier™ Release Notes



MATLAB® & SIMULINK®

MathWorks®

How to Contact MathWorks

Latest news: www.mathworks.com

Sales and services: www.mathworks.com/sales_and_services

User community: www.mathworks.com/matlabcentral

Technical support: www.mathworks.com/support/contact_us

Phone: 508-647-7000

The MathWorks, Inc.
3 Apple Hill Drive
Natick, MA 01760-2098

HDL Verifier™ Release Notes

Contents

R2017b

R2017a

R2016b

R2016a

R2015b

R2015a

R2014b

R2014a

R2013b

R2013a

R2012b

R2012a

R2011b

R2011a

R2010b

R2010a

R2009b

R2017b

Version: 5.3

New Features

Bug Fixes

SystemVerilog DPI Custom Port Widths: Generate SystemVerilog ports with bit widths that match non-byte-aligned fixed-point widths

You can now generate fixed-point ports as bit or logic vector types rather than C-compatible types. Use these types to generate non-byte-aligned port sizes and port sizes greater than 64 bits.

On the **SystemVerilog DPI** parameters pane, set **Fixed-point data type** to `Bit Vector` or `Logic Vector`. For example, for a 20-bit signed fixed-point port, the generated SystemVerilog port is `bit signed [19:0]` or `logic signed [19:0]`.

You can generate SystemVerilog DPI components with these types from MATLAB® or Simulink®. These types are also supported for tunable parameters and test-point signals. See “Supported Simulink Data Types”.

Additional FPGA-in-the-Loop Board Support: Simulate with Xilinx Zynq UltraScale+ MPSoC ZCU102 Evaluation Kit

FPGA family additions:

- Xilinx® Zynq® UltraScale+™ FPGA

FPGA board additions:

- Xilinx Zynq UltraScale+ MPSoC ZCU102 Evaluation Kit

This board is supported for FPGA-in-the-loop using a JTAG connection.

Additional FPGA-in-the-Loop Board Support: Simulate with Terasic Atlas-SoC Kit / DE0-Nano-SoC Kit, and with KCU105 or VCU108 over Ethernet

This release adds FPGA-in-the-loop support for the Terasic Atlas-SoC Kit / DE0-Nano-SoC Kit using a JTAG connection.

It also adds Ethernet connection support for:

- Xilinx Virtex® UltraScale™ FPGA VCU108 Evaluation Kit

- Xilinx Kintex® UltraScale FPGA KCU105 Evaluation Kit

SystemVerilog DPI asynchronous reset

The asynchronous reset on the SystemVerilog DPI component now updates the output ports to reflect the reset values of internal states. This update applies to SystemVerilog DPI components generated from MATLAB or Simulink.

SystemVerilog DPI component generation maintains data type inheritance in model

In previous releases, if **Hardware Implementation > Device vendor** was not set to `Custom Processor`, generating a SystemVerilog DPI component could change the data type inheritance in your model. In R2017b, SystemVerilog component generation is independent of your **Hardware Implementation** settings, and simulation data type inheritance is not affected.

TLM component generation periodic execution and signal port support

You can now set the generated TLM component to run as a time-periodic thread. Previously, the TLM component executed only when the inputs changed. On the **TLM Processing** parameters pane, set **Algorithm Step Function Execution** to `Periodic SystemC Thread`.

In addition, you can now map a port of the generated TLM component to the `sc_signal` type, by specifying `<spirit:wire>` in the IP-XACT file. See “Prepare IP-XACT File for Import”.

Updates to supported software

Supported versions of FPGA tools added this release:

- Intel® Quartus® Prime 16.1
- Xilinx Vivado® 2016.4
- Xilinx ISE 14.7

For a full list of supported software, see “Supported EDA Tools and Hardware”.

Altera FPGA Boards support package renamed

The HDL Verifier Support Package for Altera® FPGA Boards has been renamed to HDL Verifier Support Package for Intel FPGA Boards.

R2017a

Version: 5.2

New Features

Bug Fixes

FPGA Data Capture: Probe internal FPGA signals to analyze in MATLAB or Simulink

Specify signals from your HDL files to capture and return to MATLAB or Simulink. To capture the signals, HDL Verifier generates an IP core that you must integrate into your HDL project and deploy to the FPGA along with the rest of your design.

For Simulink, HDL Verifier generates a block that has output ports corresponding to the signals you captured. For MATLAB, HDL Verifier generates an app to guide you through capturing data, or a System object™ for programmatic data capture.

To use this feature, you must install the HDL Verifier Support Package for Xilinx or Altera FPGA boards. See HDL Verifier Supported Hardware.

HDL Code Coverage: Activate HDL simulator code coverage in generated test benches

When you generate a test bench, you can now add code coverage of the generated HDL code, including the test bench. The coder generates a build-and-run script that includes the flags for code coverage.

- Simulink Configuration Parameters — On the **HDL Code Generation > Test Bench** pane, select the **HDL code coverage** check box and specify your HDL simulator.
- MATLAB Command Line — When you call `makehdl` or `makehdltb`, set the `HDLCodeCoverage` property to `'on'`. Set the `SimulationTool` property to `'Mentor Graphics Modelsim'` or `'Cadence Incisive'`. Code coverage is not supported for other simulators.
- Workflow Advisor — In step 3.1.4, select the **HDL code coverage** check box and specify your HDL simulator. Then in step 3.2, select **Generate testbench**.

To use this feature, you must have an HDL Coder™ license. See Generate Test Bench and Enable Code Coverage Using the HDL Workflow Advisor.

MATLAB Based AXI Master: Interactively read and write AXI4 and AXI4-Lite signals on your FPGA

Develop an FPGA design that uses AXI4 and AXI4-Lite registers without the need for an embedded processor. Use MATLAB to read and write the generated AXI4 and AXI4-Lite registers while you run the FPGA design on the board. To use this feature:

1. Include the JTAG AXI Master IP in your FPGA project or HDL Coder reference design. This IP can communicate with MATLAB.
2. Deploy the design to your board.
3. Use MATLAB to read and write the generated AXI4 and AXI4-Lite registers in the FPGA design.

See IP Core Generation Workflow without an Embedded ARM Processor: Xilinx Kintex-7 KC705, or IP Core Generation Workflow without an Embedded ARM Processor: Arrow DECA MAX 10 FPGA Evaluation Kit.

To use this feature, you must install the HDL Verifier Support Package for Xilinx or Altera FPGA boards. See HDL Verifier Supported Hardware.

SystemVerilog DPI Test Bench Generation: Speed up test bench generation from Simulink models with large data sets

Reduce test bench generation and simulation time, especially when using large data sets.

- Simulink Configuration Parameters — On the **HDL Code Generation > Test Bench** pane, select **SystemVerilog DPI test bench**, and specify your HDL simulator.
- Workflow Advisor — In step 3.1.4, select **SystemVerilog DPI test bench**, and specify your HDL simulator. Then in step 3.2, select **Generate testbench**.

The coder generates a DPI component for your entire Simulink model, including your DUT and data sources. Your entire model must support C code generation with Simulink Coder™. The coder generates a SystemVerilog test bench that compares the output of the DPI component with the output of the HDL implementation of your DUT. The tool also generates a build script for your simulator. The supported simulators are:

- `Mentor Graphics Modelsim`
- `Cadence Incisive`

- `Vivado`
- `VCS`

To use this feature, you must have an HDL Coder license and a Simulink Coder license.

See Verify HDL Design With Large Dataset Using SystemVerilog DPI Test Bench.

Native Floating-Point Test Bench: Generate SystemVerilog DPI, cosimulation, and FPGA-in-the-loop test benches that have single-precision data types (requires HDL Coder)

When you generate a SystemVerilog DPI component, a cosimulation model or System object, or an FPGA-in-the-loop block or System object from a DUT that has `single` data type ports, the coder generates the test bench using the HDL Coder native floating-point library.

To use this feature, you must have an HDL Coder license.

See Verify the Generated Code from Native Floating-Point.

Additional FPGA Board Support: Perform FPGA-in-the-loop simulation with Xilinx Virtex UltraScale family boards

FPGA family additions

- Xilinx Virtex UltraScale FPGA

FPGA board additions:

- Xilinx Virtex UltraScale FPGA VCU108 Evaluation Kit (JTAG connection only)
- Xilinx Virtex-7 VC709 Evaluation Board (JTAG and PCI Express® connection)
- Digilent® Arty Board, or Xilinx Artix®-7 FPGA Development Board (JTAG connection only)

Generate SystemVerilog DPI components for models with complex ports in MATLAB

In the generated SystemVerilog component, the coder flattens complex signals into real and imaginary parts . See Supported MATLAB Data Types.

Updates to supported software

This section lists the supported versions added in the current release. For a full list of supported software, see Supported EDA Tools and Hardware.

HDL Cosimulation

- Cadence Incisive® 15.2
- Mentor Graphics® ModelSim® 10.5b

FPGA Verification

- Altera Quartus Prime 16.0
- Xilinx Vivado 2016.2

R2016b

Version: 5.1

New Features

Bug Fixes

FPGA-in-the-Loop for Control Applications: Return a larger output data size when using an overclocking factor

The maximum output data size of your FIL block is no longer reduced by using an overclocking factor. The output data size of your FIL block, rounded up to the nearest byte, can be up to 1467 bytes, independent of the overclocking factor.

FPGA-in-the-Loop Custom Clock Speed: Specify the FPGA system clock frequency in the FIL Wizard

In the **FIL Options** section of the FPGA-in-the-Loop Wizard, enter your **FPGA system clock frequency** in MHz.

The default clock frequency remains 25 MHz.

Multirate SystemVerilog DPI Components: Generate multirate test benches to verify that your generated component matches Simulink behavior

You can now generate a test bench for a SystemVerilog DPI component that has multirate interfaces. The generated test bench includes a timing controller that assists the stimulus and checker modules to align data at different rates.

Logic Analyzer: Visualize, measure, and analyze transitions and states over time for Simulink signals (requires DSP System Toolbox)

The Logic Analyzer visualization tool enables you to view the transitions of signals. You can use the **Logic Analyzer** to:

- Debug and analyze models
- Trace and correlate many signals simultaneously
- Detect and analyze timing violations
- Trace system execution

See Inspect and Measure Transitions Using the Logic Analyzer to explore some of its key functionality.

You must have a DSP System Toolbox™ license and a Simulink license to use this feature.

Generate SystemVerilog DPI test bench for Xilinx Vivado

When you call the `makehdltb` function, set the `GenerateSVDPITestBench` property to `'Vivado Simulator'` to create a test bench and script for the Xilinx Vivado simulator.

You must have an HDL Coder license and a Simulink Coder license to use this feature.

Cross-platform support for TLM component generation

You can generate your TLM component on any host machine for simulation on a Windows® or Linux® target machine. In the **TLM Compilation** parameter group, specify the **Operating System** of the target machine. By default, the tool generates code for the same operating system as the host machine.

Generate unmapped IP-XACT registers for TLM component

When you generate a TLM component using an IP-XACT memory map, you can now include all registers in the memory map in the generated component. Previously, the component included only registers mapped to Simulink ports and parameters, indicated with the `MWMap` tag in the IP-XACT file.

To include the unmapped registers, in the **TLM Mapping** parameter group, select the **Generate code for unmapped IP-XACT registers/bitfields** check box.

Additional FPGA Board Support: Perform FPGA-in-the-loop simulation with Altera Arria 10 family boards

- FPGA family additions:
  - Altera Arria® 10 FPGA
- FPGA board additions:
  - Altera Arria 10 SoC development kit

Updates to supported software

This section lists the supported versions added in the current release. For a full list of supported software, see Supported EDA Tools and Hardware.

FPGA Verification

- Altera Quartus Prime 15.1
- Xilinx Vivado 2015.4

DPI Component Generation

- 32-bit Cadence Incisive

TLM Component Generation

- Visual Studio® 2013

R2016a

Version: 5.0

New Features

Bug Fixes

Compatibility Considerations

PCI Express FPGA-in-the-Loop: Perform FIL simulation on selected Xilinx and Altera development boards

Perform FPGA-in-the-loop simulation on selected Xilinx and Altera FPGAs using a PCI Express connection (Windows 64 only). You must have the HDL Verifier Support Package for Xilinx or Altera FPGA boards. When you install the FPGA board support package, it also installs the PCIe driver for your board.

Board Support

HDL Verifier supports FIL over a PCIe connection on these boards:

- Xilinx Virtex-7 VC707 development board
- Xilinx Kintex-7 KC705 development board
- Altera Cyclone® V GT development kit
- Altera DSP development kit, Stratix® V edition

Software Requirements

- Xilinx Vivado 2015.2
- Altera Quartus II 15.0

Faster Test Bench Generation and HDL Simulation: Generate SystemVerilog DPI test benches for large data sets from HDL Coder

Reduce test bench generation and simulation time, especially when using large data sets. To use this feature, when you call the `makehdltb` function in HDL Coder, set the `GenerateSVDPITestBench` property. The coder generates a DPI component for your entire Simulink model, including your DUT and data sources. Your entire model must support C code generation with Simulink Coder. The coder generates a SystemVerilog test bench that compares the output of the DPI component with the output of the HDL implementation of your DUT. The tool also generates a build script for your simulator. You can specify '`ModelSim`', '`VCS`', or '`Incisive`'.

```
makehdltb(gcb,'GenerateSVDPITestBench','ModelSim','GenerateHDLTestbench','off')
```

You must have an HDL Coder license and a Simulink Coder license to use this feature.

Expanded Data Type Support in SystemVerilog DPI: Generate SystemVerilog DPI components for models that have buses, structures, or complex signals as I/O

For a list of supported data types for SystemVerilog DPI component generation, see:

- Supported Simulink Data Types
- Supported MATLAB Data Types

Additional FPGA Board Support: Perform FPGA-in-the-loop simulation with Xilinx Kintex UltraScale and Altera MAX 10 family boards

- FPGA family additions
  - Xilinx Kintex UltraScale
  - Altera MAX® 10
- FPGA board additions
  - Xilinx Kintex UltraScale FPGA KCU105 Evaluation Kit
  - Arrow® MAX 10 DECA

For a full list of supported boards, see Supported EDA Tools and Hardware.

Guided Hardware Setup for New FPGA Boards

When you download an FPGA board support package for use with HDL Verifier, the installer guides you through setting up your hardware with an Ethernet, JTAG, or PCIe connection. You can test and verify the connection before you perform FPGA-in-the-loop simulation. Doing so can eliminate problems relating to equipment and connectivity issues. Then you can focus on your algorithm and FIL performance. See Guided Hardware Setup. This feature is supported for Windows and Linux operating systems.

Updates to supported software

The following topics list the supported versions added in the current release. For a full list of supported software, see Supported EDA Tools and Hardware.

FPGA Verification

- Altera Quartus II 15.0
- Xilinx Vivado 2015.2

TLM Component Generation

- Visual Studio 2013
- Windows 7.1 SDK

Functions and Function Elements Being Removed

These TLM component generation options will be removed in a future release.

Function or Function Element Name	Use This Instead	Compatibility Considerations
Enable temporal decoupling for loosely-timed simulation	Use default timing.	Will warn in a future release.
Enable payload buffering	If you need a buffer, manually add it in your SystemC test environment.	Will warn in a future release.

R2015b

Version: 4.7

New Features

Bug Fixes

Compatibility Considerations

SystemVerilog DPI Component Test Points: Access the internal signals of a component from the test bench

In previous releases, only the input and output signals of a DPI component were visible. You can now access internal signals of a Simulink model in your HDL simulator.

See SystemVerilog DPI Component Test Point Access and Getting Started with SystemVerilog DPI Component Generation.

SystemC Modeling Library (SCML) Wrapper: Generate SCML as part of TLM component

SystemC Modeling Library is an add-on library for modeling TLM interfaces. SCML-instrumented components allow integration with Synopsys® simulation and verification tools. In the **Code Generation** dialog, on the **TLM Mapping** tab, select **Defined by imported IP-XACT file**. Then select the **Use SCML to generate the memory map** checkbox and specify an IP-XACT file for the component.

You can specify your SCML library paths on the **TLM Compilation** tab.

See Implement Memory Map with SCML.

TLM Generator: IP-XACT field support

In your IP-XACT file, you can specify more than one `field` per `register`. The generator uses the `bitWidth` and `bitOffset` tags to construct the register fields in the TLM component.

Updates to supported software

The following topics list the supported versions added in the current release. For a full list of supported software, see Supported EDA Tools and Hardware.

FPGA Verification

- Quartus II 14.0
- Xilinx Vivado 2014.4

TLM Component Generation

- SCML 2.2

Removed support for BEEcube miniBEE hardware

HDL Verifier Support Package for BEEcube® miniBEE® Platform has been removed and is no longer available.

Compatibility Considerations

In MATLAB R2015b or later, if you run a Simulink model or MATLAB code that contains BEEcube miniBEE hardware support, you will get an error

R2015a

Version: 4.6

New Features

Bug Fixes

FPGA-in-the-loop through JTAG for Xilinx boards

Perform FPGA-in-the-loop simulation on Xilinx FPGAs using a Digilent JTAG connection.

Board Support

FIL can support any Xilinx FPGA board through a Digilent JTAG connection, as long as the board uses:

- An FPGA device in the supported Xilinx FPGA family: Virtex 7, Kintex 7, Artix 7, or Zynq 7000.
- A Digilent download cable. If your board has a standard Xilinx 14-pin JTAG connector, you can obtain the HS2 cable from Digilent.

As of this release, all Xilinx 7-series FPGA boards that HDL Verifier supports directly can perform simulation through an on-board Digilent JTAG cable. To add more boards that support FPGA devices with JTAG connections, use the FPGA Board Manager.

For more information on JTAG with FIL, see JTAG Connection.

Software Requirements

- For Windows operating systems: Xilinx Vivado 2014.2. The Vivado executable directory must be on the system path.
- For Linux operating systems: Xilinx Vivado 2014.2 and Digilent Adept2.

FPGA-in-the-Loop support for rapid accelerator mode in Simulink

The FIL Simulation block now supports simulation using Rapid Accelerator mode. Before beginning cosimulation, in the model window, select **Simulation > Rapid Accelerator Mode**. See the FIL Simulation block reference for more information about this feature and block settings.

DPI-C enhancements, including multiple-instance support and integration with build toolchain

Multiple Instance Support

With the current release, multiple instances of a generated SystemVerilog DPI-C (for both Simulink and MATLAB) are supported in the HDL simulator.

Build Toolchain Integration

With build toolchain integration, you no longer have to manually build the shared library through the command-line interface. In addition, you can now target HDL simulators running on Linux from Simulink software that is running on Windows.

To specify the toolchain you want to use for code generation and (optionally) compilation:

1. In Simulink, open **Configuration Parameters**. Specify one of the supported SystemVerilog DPI-C targets.
2. From the **Build Process** list, select your toolchain.
3. Optionally, under **Build Configuration**, select any flags you want for compilation.

IP-XACT support for TLM

- Import IP-XACT files for memory map and nonmemory map TLM generation.

  To specify the files you want to import:

  1. In Simulink, select **Simulation** > **Model Configuration Parameters**.
  2. In the **TLM Mapping** pane, select **Defined by imported IP-XACT file**.
  3. When prompted, provide the full path to the IP-XACT file.
- Export IP-XACT file with generated TLM component.

  The TLM generator generates the IP-XACT file automatically and export it to the same folder as the project makefile.

Additional FPGA-in-the-loop board support

- FPGA family additions

  - Xilinx Zynq-7000
  - Xilinx Artix-7
  - Altera Arria V
- FPGA board additions
  - Digilent Nexys™4 Artix-7 FPGA board
  - Xilinx Zynq-7000 ZC702 development board
  - Xilinx Zynq-7000 ZC706 development board
  - Zedboard
  - Digilent ZYBO™ Zynq-7000 development board
  - Altera Arria V starter kit
  - Altera Arria V SoC development kit

Process improvement for SystemVerilog DPI-C generation

When selecting options for SystemVerilog DPI-C generation, you no longer have to specify the location of `svdpi.h` and `libvsim.lib` files.

Delay propagation and extra control signals eliminated from generated SystemVerilog code

DPI component generation now provides a SystemVerilog template for AMS workflows that eliminate delay propagation and extra control signals.

TLM generation updates

With the current release, TLM Generation from HDL Verifier provides:

- Bus and structure support
- SystemC 2.3.1 support
- Multiple instance support

R2014b

Version: 4.5

New Features

Bug Fixes

SystemVerilog DPI-C component generation based on MATLAB Coder

With the current release, you can export a MATLAB design with direct programming interface (DPI) for Verilog® or SystemVerilog simulation. With this feature, you can wrap generated C code with a DPI wrapper that communicates with a SystemVerilog thin interface function in a SystemVerilog simulation.

For more on this feature, see DPI Component Generation for MATLAB Function.

Note You must have a MATLAB Coder license to use this feature.

SystemVerilog DPI-C component generation based on Simulink Coder

To generate a SystemVerilog component using Simulink, you no longer require Embedded Coder®. In Code Generation, select `systemverilog_dpi_grt.tlc` as the **System target file**.

Xilinx Vivado support for FPGA-in-the-Loop

With this release, you can use Xilinx Vivado for FIL simulation. Xilinx Vivado supports 7-series and newer FPGA families. HDL Verifier supports Xilinx Vivado version 2013.4.

In the FIL simulation workflow, specify Vivado when you Set Up Hardware and Hardware Tools:

```
hdlsetuptoolpath('ToolName','Vivado','ToolPath','c:\HDLTools\Vivado\2013.4-mw-0\Win')
```

SGMII interface support for FPGA-in-the-Loop in Xilinx Virtex-7 FPGAs

With the current release, you can use the Virtex-7 VC707 Development Board with SGMII interface for FIL simulation. This board requires Xilinx Vivado version 2013.4.

You can download support for Virtex-7 VC707 in the HDL Verifier Support Package for Xilinx FPGA Boards.

Additional FPGA-in-the-Loop board support: Xilinx Virtex-7 FPGA VC707 Evaluation Kit, Arrow SoC Kit Evaluation Board, Altera Cyclone V GT FPGA Development Kit

HDL Verifier FPGA-in-the-Loop verification has added support for the following FPGA boards:

- Xilinx Virtex-7 FPGA VC707 Evaluation Kit
- Arrow SoC Kit Evaluation Board
- Altera Cyclone V GT FPGA Development Kit

For a full list of supported boards, see Supported EDA Tools and Hardware.

Updates to supported software

The following topics list the supported versions added in the current release. For a full list of supported software, see Supported EDA Tools and Hardware.

HDL Cosimulation

- Incisive® 13.2 p002
- Questa®Sim 10.3

FPGA Verification

- Xilinx 14.7
- Quartus II 13.1
- Xilinx Vivado 2013.4

TLM Component Generation

- Visual Studio: VS2012
- SystemC 2.3.1 (TLM included)

Documentation installation with hardware support package

Starting in R2014b, each hardware support package installs with its own documentation. See HDL Verifier Supported Hardware for a list of support packages available for HDL Verifier, with links to documentation.

R2014a

Version: 4.4

New Features

Bug Fixes

FPGA-in-the-Loop over JTAG for Altera FPGAs

Perform FPGA-in-the-Loop simulation on Altera FPGAs using a JTAG connection.

- Supported FPGA devices: Any Altera FPGA board within the supported FPGA family, for example:
  - Cyclone III, IV, V, and V SoC
  - Arria II
  - Stratix IV and V
  - Additional boards within the Supported FPGA Devices for FIL Simulation families can be custom added with the FPGA Board Manager.
- Hardware:
  - Altera FPGA boards
  - USB Blaster I or USB Blaster II download cable
- Software:
  - Windows: Quartus II 12.1 or higher version; Quartus II executable folder must be on system path
  - Linux: Quartus II 13.0sp1 with a patch, or Quartus II 13.1 (Quartus II library folder must be on LD_LIBRARY_PATH *before* starting MATLAB); only 64-bit Quartus are supported
  - Installation of USB Blaster I or II driver

Parameter Tuning for Generated TLM Component

The tunable parameters register allows you to make adjustments to the TLM component before or during simulation. You set this parameter in the Configuration Parameters dialog, in the **TLM Mapping** tab. See Select TLM Mapping Options.

Multiple Socket Control for Generated TLM Component

You can choose to have a single, combined TLM socket for input data, output data, and control, or you can choose three separate TLM sockets for input data, output data, and control, so that you can connect the sockets to different buses. Set this parameter in the

Configuration Parameters dialog, in the **TLM Mapping** tab. See Select TLM Mapping Options.

FPGA-in-the-Loop support for Altera Cyclone V SoC FPGA boards

You can use Altera Cyclone V SoC FPGA boards for FIL simulation with a JTAG connection.

Updates to supported software and hardware

Software updates

HDL Cosimulation

- ModelSim PE 10.2c
- QuestaSim 10.2c
- Incisive 13.10-s006

FPGA Verification

- Xilinx ISE 14.6
- Altera Quartus II 13.0sp1

Hardware support updates

- Altera Cyclone V SoC development board

For a list of supported boards and device families, see Supported FPGA Devices. To add a custom board for use with FIL, see Create Custom FPGA Board Definition. For instructions on downloading the FPGA board support packages with the Support Package Installer, see Support Packages and Support Package Installer.

R2013b

Version: 4.3

New Features

Bug Fixes

Compatibility Considerations

SystemVerilog DPI component generation from Simulink

In R2013b, you can export a Simulink subsystem with a direct programming interface (DPI) for Verilog or SystemVerilog Simulation. With this feature, you can wrap generated C code with a DPI wrapper that communicates with a SystemVerilog thin interface function in a SystemVerilog simulation.

This feature is available in the Model Configuration Parameters dialog box:

1. In Code Generation, select `systemverilog_dpi_ert.tlc` as the **System target file**.
2. Expand Code Generation, and select DPI Generator.
3. Specify the DPI include path, and indicate whether or not you want to generate a test bench.
4. Click **OK** to accept your choices and exit the dialog box or **Apply** to continue making changes.

Note You must have an Embedded Coder license to use this feature.

BEEcube miniBEE FPGA-in-the-Loop (FIL) support package

The BEEcube miniBEE hardware platform is available for FIL simulation as an HDL Verifier support package. The miniBEE support package requires you to install a PCI Express connection for host-board communications.

Note The BEEcube miniBEE hardware platform support package works only on the CentOS that ships with miniBEE.

For instructions on downloading HDL Verifier support packages with the Support Package Installer, see FPGA Board Support Packages for FIL.

After the Support Package Installer has started, select `BEEcube miniBEE Platform` at **Select a support package to install** pane in the installer GUI. For more information about using this support package with FIL, see Support Package for BEEcube miniBEE Hardware Platform.

Additional FPGA board support for FIL, including Xilinx KC705 and Altera DSP Development Kit, Stratix V edition

Several FPGA boards have been added to the HDL Verifier FPGA board support packages, including Xilinx KC705 and Altera DSP Development Kit, Stratix V edition.

For a full list of boards added, see “Updates to supported software and hardware” on page 9-4. For instructions on downloading the FPGA board support packages with the Support Package Installer, see Support Packages and Support Package Installer.

Floating-point data type for cosimulation and FIL blocks

Double and single data types on the DUT interface are supported for HDL cosimulation and FIL test bench generation. This feature is also available for System objects.

HDL file compilation ordering in Cosimulation Wizard

VHDL® files are automatically sorted into the right compilation order in the HDL Cosimulation Wizard, saving you time and, in some cases, errors. You can add files in any order you choose. You can also still manually arrange the HDL files by using the **Up** and **Down** buttons.

Shared memory connection in Cosimulation Wizard

Shared memory is an available connection method in the HDL Cosimulation Wizard. Use shared memory communication if your firewall policy does not allow TCP/IP socket communication. In the Simulation Options pane, select `Shared Memory` for the **Connection method**.

SGMII board support for FPGA-in-the-Loop simulation

SGMII support has been added to FPGA-in-the-Loop simulation. You can now perform FIL simulation with Altera Stratix IV and Stratix V FPGA boards that require an SGMII I/O interface.

Floating point for FIL and HDL cosimulation test bench generation

With the R2013b release, HDL Verifier supports double and single data types on the DUT interface for test bench generation using HDL Coder HDL Workflow Advisor for Simulink.

Updates to supported software and hardware

Software updates

HDL Cosimulation

- ModelSim SE 10.1c

FPGA Verification

- Xilinx ISE 14.4
- Altera Quartus II 12.1sp1

Hardware support updates

Device Family Support Additions

- Cyclone V

Board Support Additions

- Altera Stratix IV GX FPGA development kit
- Altera Cyclone V GX FPGA development kit
- Altera DSP Development Kit, Stratix V edition
- BeMicro SDK
- Xilinx Kintex-7 KC705 board

For a list of supported boards and device families, see Supported FPGA Devices. To add a custom board for use with FIL, see Create Custom FPGA Board Definition. For instructions on downloading the FPGA board support packages with the Support Package Installer, see Support Packages and Support Package Installer.

Functions and Function Elements Being Removed

Function or Function Element Name	What Happens When You Use The Function or Element?	Use This Instead	Compatibility Considerations
`FrameBasedProcessing` property	Warns	Sample mode or frame mode is automatically detected based on the size of the inputs during the step method execution.	In a future release, the use of any scripts containing this property will cause an error.
FPGA Automation pane in the Generate HDL dialog box	Errors		The FPGA Automation pane has been removed.

R2013a

FPGA-in-the-loop test bench generation through HDL Workflow Advisor for MATLAB

With the MATLAB Coder Workflow Advisor, the HDL Verification step includes automation for the following workflow:

- Verify with FPGA-in-the-Loop: Create the FPGA programming file and test bench, and, optionally, download it to your selected development board.

  Note You *do* require an HDL Verifier license to use this feature.

HDL cosimulation test bench generation through HDL Workflow Advisor for MATLAB

With the MATLAB Coder Workflow Advisor, the HDL Verification step includes automation for the following workflows:

- Verify with HDL Test Bench: Create a standalone test bench. You can choose to simulate using ModelSim or Incisive with a vector file created by the Workflow Advisor.

  Note You do not require an HDL Verifier license to use this feature.

- Verify with Cosimulation: Cosimulate the device under test (DUT) in ModelSim or Incisive with the test bench in MATLAB.

  Note You *do* require an HDL Verifier license to use this feature.

Transaction Level Model generation using Simulink Coder

You can generate Transaction Level Models with either a Simulink Coder license *or* an Embedded Coder license. You are not required to have both product licenses.

In Model Configuration Parameters, under **Code Generation**, follow these guidelines for selecting the correct system target file:

- With Simulink Coder, select: `tlmgenerator_grt.tlc`

- With Embedded Coder, select: `tlmgenerator_ert.tlc`

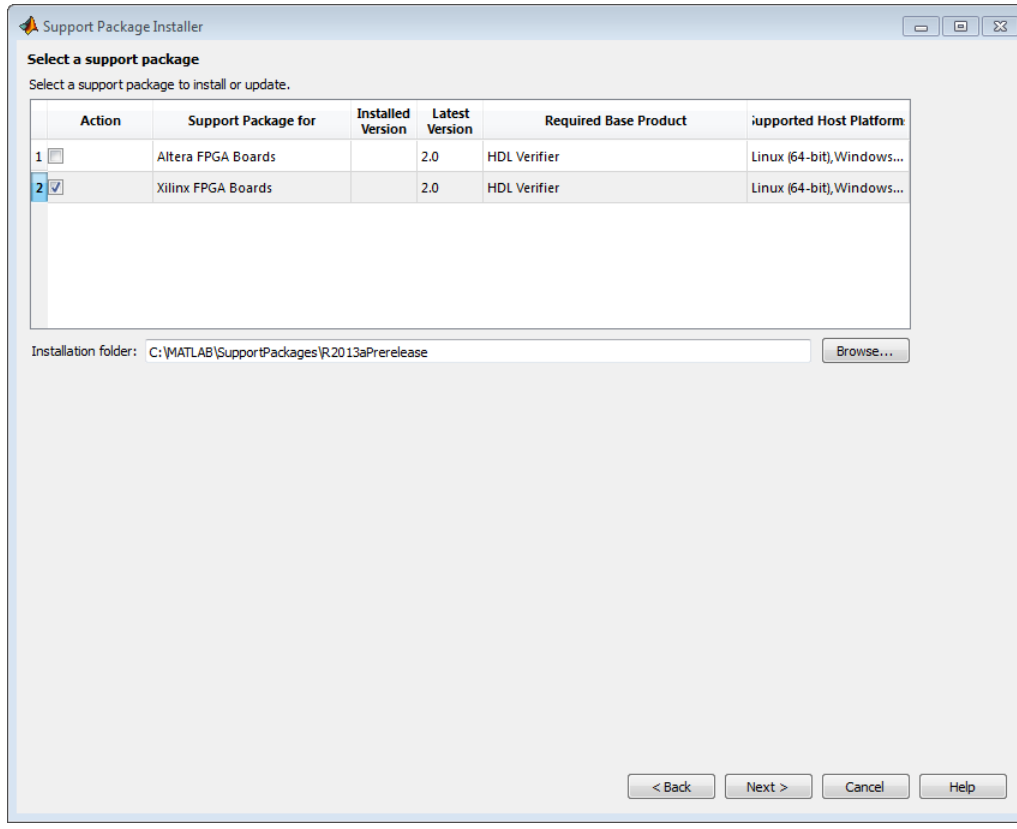
Compatibility Considerations

You no longer require Embedded Coder to generate Transaction Level Models with HDL Verifier.

Support Package for FPGA-in-the-Loop

All FPGA boards supported by FPGA-in-the-Loop are now available via download with the Support Package Installer. You can choose to download all supported Altera boards or all supported Xilinx boards, or both.

When you are using the FPGA Board Manager, select **Get More Boards** to start the Support Package Installer with the FPGA board download packages already displayed. You can also access the installer through the FIL Wizard `Get more boards` option. The installer guides you through selecting and installing the board support package. After the installer has completed the download, you can access any supported board through the FPGA Board Manager or the FIL Wizard.



Code Generation for FIL Simulation Block

With Release R2013a, you can generate code for the FIL Simulation block.

Updates to supported software and boards

- "Software updates" on page 10-5
- "Board additions" on page 10-5

Software updates

FPGA Verification

- Xilinx ISE 14.2
- Altera Quartus™ II 12.0

TLM Generation

- Compilers:
  - Visual Studio: VS2005, VS2008, VS2010
  - gcc 4.4.6
- SystemC:
  - SystemC 2.3.0 (TLM included)

Board additions

- Altera DSP Stratix V

For a list of supported boards and device families, see Supported FPGA Devices. To add a custom board for use with FIL, see Create Custom FPGA Board Definition.

Compatibility Considerations

Xilinx does not ship the Digilent plugin with ISE 14.2. To get the plugin, see the Digilent plugin and related software download page on the Digilent web site.

HDL Verifier No Longer Supports Legacy FIL Programming Files

FPGA-in-the-Loop FPGA programming files generated using HDL Verifier versions older than 4.2 (current release) are not compatible with the current version of software.

Compatibility Considerations

FPGA programming files that were generated with version 4.1 or earlier of HDL Verifier are unusable with the current version of software (R2013a). You must regenerate these programming files using the FIL Wizard or HDL Workflow Advisor with the current release.

Functions and Function Elements Being Removed

Function or Function Element Name	What Happens When You Use The Function or Element?	Use This Instead	Compatibility Considerations
`FrameBasedProcessing` property	Warns	Sample mode or frame mode is automatically detected based on the size of the inputs during the step method execution.	In a future release, the use of any scripts containing this property will cause an error.
FPGA Automation pane in the Generate HDL dialog box	Warns		In a future release, the FPGA Automation pane will be removed.

R2012b

Version: 4.1

New Features

Bug Fixes

Custom board APIs for FPGA-in-the-loop

New FPGA Board Manager allows you to add custom board information so that you can use FIL simulation with an FPGA board that is not one of the pre-registered boards. See Create Custom FPGA Board Definition.

System object for FPGA-in-the-Loop

FIL System object for FIL simulation between FPGA and MATLAB. The`hdlverifier.FILSimulation` System object can be generated only with the FIL Wizard. See FIL Wizard: Generate FIL System Object.

100 Base-T Ethernet support for FPGA-in-the-loop block

HDL Verifier supports FPGA boards that have 100 Mbit/sec Ethernet PHY. The software will automatically detect the configuration of the PHY chip and employ the appropriate interface in the FPGA as necessary for FIL simulation.

Automatic verification with cosimulation using HDL Coder

With the HDL Coder HDL Workflow Advisor, you can automatically verify using your Simulink test bench with the new verification step **Run Cosimulation TestBench**. During verification, the HDL Workflow Advisor and HDL Verifier verify the generated HDL using cosimulation between the HDL Simulator and the Simulink test bench. See Automatic Verification.

Updates to supported software and boards

Software updates

- ModelSim 10.1a, 10.0c, and 6.6d
- Cadence Incisive 11.10-s005
- Xilinx ISE 13.4

Board additions

- Altera Nios II Embedded Evaluation Kit, Cyclone III Edition

For a full list of preregistered boards, see Supported FPGA Devices for FIL Simulation. To add a custom board for use with FIL, see Create Custom FPGA Board Definition.

R2012a

Version: 4.0

New Features

Bug Fixes

Compatibility Considerations

EDA Simulator Link Is Now HDL Verifier

Effective R2012a, EDA Simulator Link is now HDL Verifier.

FPGA-in-the-Loop for Altera Boards

FPGA-in-the-Loop now supports Altera FPGA design software and the following Altera development kits and boards:

- Altera Arria II GX FPGA development kit
- Altera Cyclone III FPGA development kit
- Altera Cyclone IV GX FPGA development kit
- Altera DE2-115 development and education board

See Required Products, Performing FPGA-in-the-Loop.

System Object for HDL Cosimulation with MATLAB, with Automatic System Object Generation

The HDL cosimulation System object provides integrated HDL cosimulation with MATLAB. When you use this workflow to cosimulate MATLAB and HDL code, you gain the following benefits of using the System object:

- Control all aspect of the cosimulation from MATLAB.
- Easily configure all test bench parameters.
- Remove need for multiple function calls.
- Create System object automatically from existing HDL code (see “Automatic System Object Generation with CosimWizard” on page 12-2).

The HDL cosimulation System object supports HDL cosimulation with both Mentor Graphics ModelSim and Cadence Incisive. You can read more about the HDL cosimulation System object and its methods and properties in the HDL Cosimulation System Objects reference page.

Automatic System Object Generation with CosimWizard

Although you can hand code an HDL cosimulation System object, you can more easily create the System object automatically using existing HDL code and the HDL

Cosimulation Wizard. This workflow also creates an HDL launch script for easier startup.

See Creating a Function, System Object, or Block.

Use of FPGA Board as Source Block with FPGA-in-the-Loop

Effective R2012a, you can use the FPGA board as the source of stimuli in Simulink. Only one output is required. This feature enables high speed generation and processing of test stimulus with results brought back to Simulink for analysis.

See the example “Algorithm Verification with FIL Source Block”.

HDL Regression Testing with Simulink Design Verifier

You can now perform successive simulation runs without restarting the HDL simulator. This enhancement allows uses of automatically generated test cases, from the original behavioral model, generated from Simulink Design Verifier™. Such regression testing can achieve complete model and HDL code coverage.

Previously, commands issued in the Tcl startup file were executed only once, when the HDL simulator was started. Now, Tcl commands in the Simulation pane of the HDL Cosimulation block are stored with the block and are issued with each new simulation run. You do not have to reissue the commands in the HDL simulator.

See the example “Generating HDL Code Coverage Using Simulink and ModelSim”.

New Examples for R2012a

- HDL Cosimulation with MATLAB System object:
  - Cosimulation Wizard for MATLAB System object
  - Verifying Viterbi Decoder Using MATLAB System object and Cadence Incisive
  - Verifying Viterbi Decoder Using MATLAB System object and Mentor Graphics ModelSim
- Accelerate Algorithm Verification with a FIL Source Block
- Generating HDL Code Coverage Using Simulink and Mentor Graphics ModelSim
- FIL Demos Updated for Altera Workflow

HDL Verifier Supported Software and System Updates

The HDL Verifier supported software updates for this release are listed in the following sections.

See Required Products for a complete listing of supported products.

HDL Cosimulation

- Support for ModelSim 10.0c added
- No change to supported Cadence Incisive versions

FPGA-in-the-Loop

- Altera Quartus II 11.0
- No change to supported Xilinx Design Suite versions

Functions and Function Elements Being Removed

Function or Function Element Name	What Happens When You Use The Function or Element?	Use This Instead	Compatibility Considerations
FPGA Automation in Configuration Parameters or Model Explorer	You cannot access this feature	Use the HDL Workflow Advisor.	Use replacement workflow.
`fpgamodelsetup`	Errors	Use the HDL Workflow Advisor	Use replacement workflow.
`makefpgaproject`	Errors	Use the HDL Workflow Advisor	Use replacement workflow.
`configuremodelsim`	Errors	`vsim`	Use replacement function.

Function or Function Element Name	What Happens When You Use The Function or Element?	Use This Instead	Compatibility Considerations
`wrapverilog`	Errors	Wrapping Verilog code is no longer required.	

R2011b

FPGA-in-the-Loop Workflow in HDL Coder HDL Workflow Advisor

FPGA-in-the-Loop (FIL) is available using the HDL Coder HDL Workflow Advisor. You can verify FPGA designs with FIL as part of the HDL Workflow Advisor workflow, which will create the FPGA programming file and download it to your selected development board. See HDL Coder documentation for details.

FPGA-in-the-Loop Updates

This release removes previous limitations and now supports:

- Arbitrary vectors. Limitations on input and output signal size in FIL have been removed.
- Variable step solvers
- ode45 solvers and model referencing

Conversion of Error and Warning Message Identifiers

For R2011b, EDA Simulator Link error and warning message identifiers have changed.

Compatibility Considerations

If you have scripts or functions that use message identifiers that changed, you must update the code to use the new identifiers. Typically, message identifiers are used to turn off specific warning messages.

For example, the `edalink:filWizard:NotString` identifier has changed to `edalink:filWizard:NotString`. If your code checks for `edalink:filWizard:NotString`, you must update it to check for `edalink:filWizard:NotString` instead.

To determine the identifier for a warning that appears at the MATLAB prompt, run the following command after you see the warning:

```
[MSG,MSGID] = lastwarn;
```

This command saves the message identifier to the variable `MSGID`.

Note Warning messages indicate a potential issue with your model or code. While you can turn off a warning, a suggested alternative is to change your model or code so it does not generate warnings.

EDA Simulator Link Supported Software and System Updates

EDA Simulator Link supported software updates for this release include:

- ModelSim SE 10.0a, 6.6d, 6.5f
- ModelSim PE 10.0a, 6.6d, 6.5f
- ModelSim DE 10.0a (Windows 32 only)
- Questa 10.0a
- Cadence® IES 10.2-s040
- Cadence IES 9.2-s014
- Cadence IUS 8.2-s009
- Xilinx ISE 13.1

See Required Products.

Functions and Function Elements Being Removed

Function or Function Element Name	What Happens When you use the Function or Element?	Use This Instead	Compatibility Considerations
FPGA hardware-in-the-loop via the EDA Link pane in Configuration Parameters or Model Explorer	Errors	`filWizard` or FIL in the HDL Workflow Advisor (HDL Coder).	Use replacement function
FPGA Automation in Configuration Parameters or Model Explorer	Warns	Use the HDL Workflow Advisor.	Use replacement workflow

Function or Function Element Name	What Happens When you use the Function or Element?	Use This Instead	Compatibility Considerations
`configuremodelsim`	Errors	`vsim`	Use replacement function
`launchDiscovery`	Errors	Support for Synopsys Discovery™ has been removed.	You must remove the `launchDiscovery` command from any scripts or programming files that contain it.
HDL Cosimulation block—Discovery	Errors	Support for Synopsys Discovery has been removed.	Remove or replace the Discovery HDL Cosimulation block in your Simulink models.

R2011a

Version: 3.3

New Features

Bug Fixes

Compatibility Considerations

FPGA-in-the-Loop Simulation

This release provides the capability for verification of FPGA designs with FPGA-in-the-Loop (FIL) simulation. The FIL Wizard, using HDL files that you provide, creates all the FPGA programming files and downloads them to a development board. It also provides you with a FIL block to insert into your existing model so that you can then run and test your FPGA implementation on the development board using Simulink. See Generating a FIL Simulation Block for details.

EDA Simulator Link tested FIL simulation with: Xilinx ISE 12.1; Supports Windows 32, Windows 64, Linux 32, Linux 64.

Multiple Cosimulation Sessions Support with Parallel Computing

You can use the EDA Simulator Link and Parallel Computing Toolbox™ products together for up to eight cosimulation sessions on local machine. Use MATLAB Distributed Computing Server™ to farm out sessions to any number of other computers or to run more than eight sessions on a local machine.

Refer to the Parallel Computing Toolbox and the MATLAB Distributed Computing Server documentation for details and examples of using parallel computing with MATLAB and Simulink.

New User Guide Section for Using HDL Instance Object with Test Bench and Component Functions

Expanded documentation helps you learn how to use the `use_instance_obj` argument for MATLAB functions `matlabcp` and `matlabtb`. You can use this feature to pass an HDL instance object to the function as an argument. In previously releases, the `iport`, `oport`, `tnext`, `tnow`, and `portinfo` arguments of the MATLAB function definition served this purpose. With this feature, `matlabcp` and `matlabtb` function callbacks get the HDL instance object passed in: to hold state, provide read/write access protection for signals, and allow you to add state as you wish.

See Writing Functions Using the HDL Instance Object for details.

EDA Cosimulation Assistant Name Change to Cosimulation Wizard

The feature name "EDA Cosimulation Assistant" has been changed to "Cosimulation Wizard". The function used to launch the wizard has changed also. The function

`edaCosimAssist` will be removed in a future release. Although `edaCosimAssist` is supported for backward compatibility, you should use function `cosimWizard` instead. See Creating a Function, System Object, or Block.

Compatibility Considerations

Replace all existing instances of `edaCosimAssist` with `cosimWizard`.

EDA Simulator Link Supported Software and System Updates

- For FPGA Automation (with Simulink or Filter Design HDL Coder™):
  - Tested with Xilinx ISE 12.1
  - Added Windows 64 support
- For HDL Cosimulation:
  - ModelSim SE 6.6c, 6.5f, 6.4g
  - ModelSim PE 6.6c, 6.5f, 6.4g
  - ModelSim DE 6.6c (Windows 32 only)
  - Questa 6.6c
  - Support for Synopsys Discovery will be removed in a future release.
- Windows 64 Support Added for TLM Component Generation

Functions and Function Elements Being Removed

Function or Function Element Name	What Happens When you use the Function or Element?	Use This Instead	Compatibility Considerations
`edaCosimAssist`	Still runs	`cosimWizard`	Replace all existing instances of `edaCosimAssist` with `cosimWizard`.

Function or Function Element Name	What Happens When you use the Function or Element?	Use This Instead	Compatibility Considerations
`launchDiscovery`	Warns		Support for Synopsys Discovery will be removed in a future release.
HDL Cosimulation block—Discovery	Warns		Support for Synopsys Discovery will be removed in a future release.
FPGA hardware-in-the-loop	Warns	FPGA-in-the-Loop	Support for FPGA HIL will be removed in a future release.

R2010b

Version: 3.2

New Features

Compatibility Considerations

HDL Cosimulation Updates

EDA Cosimulation Assistant Creates Blocks and Functions from Existing HDL Code

Get started quickly using existing HDL code and the EDA Cosimulation Assistant. This tool will guide you through the steps to create a test bench or component function for cosimulation with MATLAB or an HDL Cosimulation block for cosimulation with Simulink. See Generate HDL Cosimulation Interfaces from Existing HDL Code.

Updated Timescales Pane Offers New Options for Simulation Timescale Factoring

The updated timescale features allows you to choose when EDA Simulator Link software should calculate a timescale for you. In addition, you can make changes to the calculated timescale with an interactive GUI. See the HDL Cosimulation block reference for more information.

To VCD File Block Supports Simulation Using Rapid Accelerator Mode

To VCD File block now supports simulation using Rapid Accelerator mode. Select **Simulation > Rapid Accelerator Mode** in the model window before beginning cosimulation. See the To VCD File block reference for more information about this feature and block settings.

EDA Simulator Link Supports ModelSim DE

EDA Simulator Link software now supports ModelSim DE 6.6a (for Windows only). See Product Requirements for more information on supported products.

EDA Simulator Link Supports Cosimulation on 64-Bit Windows

This release adds support for Windows 64-bit machines. See Product Requirements for more information on supported products.

HDL Cosimulation Support for Synopsys Updated

With the release of R2010b, EDA Simulator Link software now supports VCS® MXi. The software no longer supports VCS MX.

Compatibility Considerations

Attempts to cosimulate with VCS MX will be result in errors.

Zero Value of First Output for All Signals Corrected

Previously, the first output value of all signals in the HDL code was set to zero in Simulink (unless you used direct feedthrough, which works as expected). Effective this release, the output value of all signals now pass from the HDL simulator to Simulink as expected.

Compatibility Considerations

You will no longer see zero as the first output value of all signals in Simulink. You may need to modify your code or procedures to accommodate the corrected behavior.

FPGA Automation Updates

FPGA Project Generation with MATLAB and Filter Design HDL Coder

You can create Xilinx ISE projects from Filter Design HDL Coder and MATLAB. From the Filter Design & Analysis Tool GUI, select **Targets > Generate HDL**. Select the **FPGA Automation** tab. Use context-sensitive help to assist you in setting project generation options.

The software now supports FPGA project generation with MATLAB and Filter Design HDL Coder on 32-bit Windows systems and on 32- and 64-bit Linux systems.

See FPGA Project Generation with Xilinx to get started.

Clock Module Generation Now Supports Verilog

The limitation for DCM design containing only VHDL code has been removed. You may select Verilog on the HDL Coder pane or the Generate HDL pane (in Filter Design HDL Coder) when you specify options for FPGA project generation.

TLM Generation Updates

Single Source and Sink Blocks Now Supported

This release removes the previous limitation that designs under generation must contain both inputs and outputs. Your design now may have only source blocks or only sink blocks.

New Algorithm Processing Options

Choose the type of function execution trigger you want to use in the generated TLM component—SystemC thread or callback. Use a SystemC thread for a more realistic simulation (at the expense of time) and use a callback for faster execution (at the expense of accuracy).

Temporal Decoupling Added to Generated TLM and Test Bench

This release includes complete temporal decoupling implementation with quantum for faster processing for generated TLM and test bench.

R2010a

Version: 3.1

New Features

Support for Latest Synopsys Discovery Release

EDA Simulator Link now supports the latest Synopsys release. See the requirements page on the MathWorks® Web site for specific platforms supported and detailed information about the software and hardware required to use EDA Simulator Link software with the current release.

Enable Direct Feedthrough for HDL Designs with Pure Combinational Datapaths

The HDL Cosimulation block now supports direct feedthrough, which means that the output is controlled directly by the value of an input port. The input value change propagates to the output ports in zero time, thus eliminating one output-sample delay for HDL designs with pure combinational logic datapaths. This feature eliminates the need to modify the test bench portion of Simulink to compensate for cosimulation block delay.

New Functions for HDL Simulator Client Communication

A new function, `notifyMatlabServer`, allows you to send HDL simulator event and process IDs to MATLAB server. Another new function, `waitForHdlClient`, waits to begin processing until the specified event ID is obtained or a user-specified time-out occurs.

Batch, CLI, and GUI Mode Support Added for Cosimulation with HDL Simulators

You can execute cosimulation in batch mode for background processing or CLI mode for ease in debugging.

Use Same MATLAB Function for Multiple HDL Instances

This release adds a new argument, `use_instance_obj`, to the MATLAB functions `matlabcp` and `matlabtb`. This feature replaces the `iport`, `oport`, `tnext`, `tnow`, and `portinfo` arguments of the MATLAB function definition with an HDL instance object passed to the function as an argument. With this feature, `matlabcp` and `matlabtb` function callbacks get the HDL instance object after it has passed into hold state. They also provide read/write access protection for signals and allow you to add state as you wish.

With this feature you gain the following advantages:

- You can use the same MATLAB function to represent behavior for different instances of the same module in HDL without need to create one-off wrapper functions.
- You no longer need special `portinfo` argument on first invocation.
- You no longer need to use persistent or global variables.
- You receive better feedback and protections on reading/writing of signals.
- You can use object fields to identify the instance path and whether the call comes from a component or test bench function.
- You can use the field argument to pass user-defined arguments from the `matlabcp` or `matlabtb` instantiation on the HDL side to the function callbacks.

The new argument, `-use_instance_obj`, is identical for both `matlabcp` and `matlabtb`. See the Function Reference for `matlabcp` and `matlabtb` for instructions in using this new function argument.

Generating Transaction Level Models for Use with Virtual Platforms

- Export of Simulink algorithm models as OSCI TLM 2.0 components
- Generation of standalone SystemC test bench for generated TLM 2.0 component

Limitations The design under generation must contain both inputs and outputs. Designs that only have inputs or only have outputs (sink or source blocks) are not supported in this release. If you do not include both, EDA Simulator Link displays an error message and discontinues code generation.

Specializing FPGA Implementations

Generation of Xilinx ISE projects for FPGA designs

R2009b

Version: 3.0

New Features

EDA Simulator Link DS, EDA Simulator Link IN, and EDA Simulator Link MQ Merge

As of R2009b, EDA Simulator Link DSo, EDA Simulator Link INo, and EDA Simulator Link MQo functionality are merged into a new product, EDA Simulator Link. These individual products are no longer available.