Simulink® Interface

Learn how to use Simulink® Interface in Active-HDL

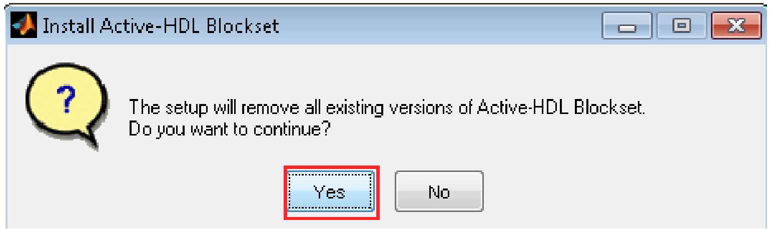
Overview

- The MathWorks' **MATLAB®/Simulink®** simulation environment provides a powerful high level mathematical modeling environment for DSP systems that can be widely used for algorithm development and verification.
- Active-HDL provides an interface to **MATLAB** and **Simulink** simulation environment, which allows co-simulation of functional blocks described by using mathematical formulas and behavioral models described by using hardware description languages.
- The **Simulink** Interface provides users with the following benefits:
    - Intuitive interface that fills the gap between HDL simulation and high level mathematical modeling environment for DSP systems
    - Displaying simulation results in both the **Simulink** environment and the **Active-HDL** waveform window
    - Automatic value conversion between **Active-HDL** and **Simulink** data types, including **Simulink** fixed-point types.
    - Advanced testbenches employing complex mathematical formulas used to stimulate unit under test
    - Integration with **Xilinx System Generator™**

Installing Active-HDL Blockset

- In order to install Simulink Interface after Active-HDL is installed, follow steps below:

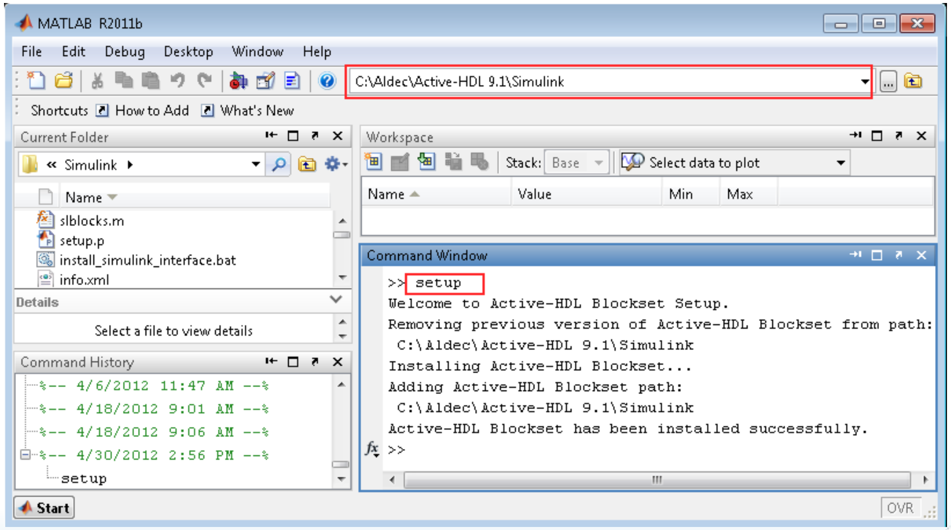
1. Start MATLAB.
2. Browse to the **$ALDEC/ Simulink subfolder.**
3. Type **setup command in the Command Window.**
4. Click Yes on pop-up warning.



Install Active-HDL Blockset

The setup will remove all existing versions of Active-HDL Blockset.
Do you want to continue?

- After the setup has finished successfully, MATLAB will display success message in console window.



MATLAB R2011b

C:\Aldec\Active-HDL 9.1\Simulink

Current Folder: « Simulink ▸

Name
slblocks.m
setup.p
install_simulink_interface.bat
info.xml

Details: Select a file to view details

Command History

```
%-- 4/6/2012 11:47 AM --%
%-- 4/18/2012 9:01 AM --%
%-- 4/18/2012 9:06 AM --%
%-- 4/30/2012 2:56 PM --%
setup
```

Command Window

```
>> setup
Welcome to Active-HDL Blockset Setup.
Removing previous version of Active-HDL Blockset from path:
 C:\Aldec\Active-HDL 9.1\Simulink
Installing Active-HDL Blockset...
Adding Active-HDL Blockset path:
 C:\Aldec\Active-HDL 9.1\Simulink
Active-HDL Blockset has been installed successfully.
>>
```

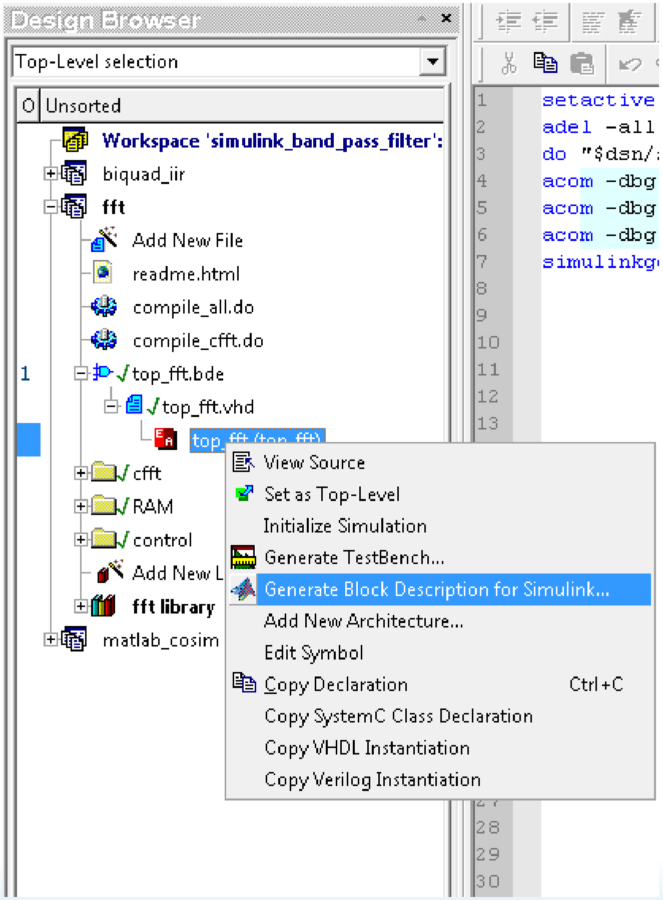
Generating Block Description for Simulink

- To start the co-simulation process in **MATLAB** environment, first you need to create **Block Description** files (MATLAB M-Files) for **Simulink** and then select **units** to be co-simulated.

This step can be done by using the '**Generate Block Description for Simulink'** option:

1. Open the Active-HDL workspace (e.g. simulink_band_pass_filter.aws) collecting designs describing units to be co-simulated.
2. Compile sources.
3. Expand a source file (e.g. top_fft.bde) that contains a design unit that will be used as a black-box on the Simulink Diagram.
4. Right-click this unit (e.g. top_fft(top_fft)) and choose the **Generate Block Description for Simulink…**option

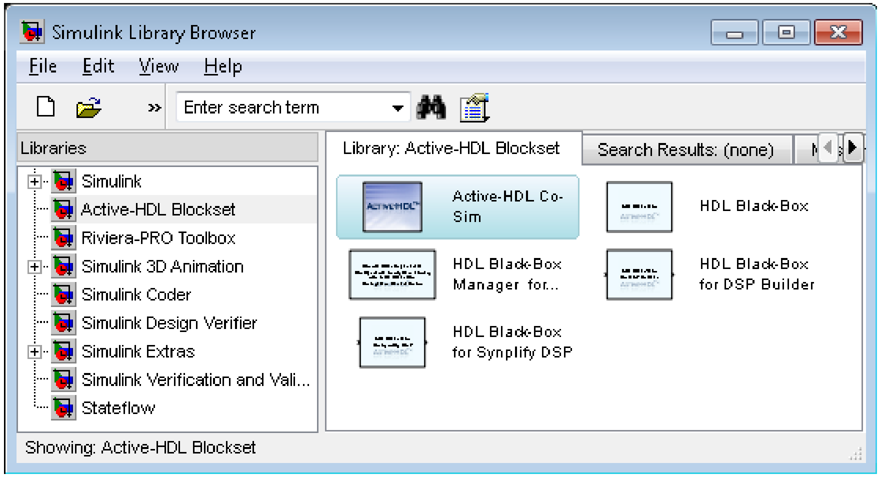
- Block Description files are generated to $WSP/ Simulink folder by default. User can change this location. Generated configuration files are dedicated for Active-HDL Blockset, common for all Active-HDL designs.



```
setactive
adel -all
do "$dsn/:
acom -dbg
acom -dbg
acom -dbg
simulinkg
```

Using Active-HDL Blockset

- The **Active-HDL Blockset** is available inside the **Simulink Library Browser**.



1. Start Simulink.
2. Choose the Simulink icon from MATLAB's main toolbar to open the Simulink Library Browser window.
3. In the left pane of Simulink Library Browser window, the Active-HDL Blockset is displayed.

The following blocks are visible in the blockset:
Active-HDL Co-Sim
HDL Black-Box
HDL Black-Box for DSP Builder
HDL Black-Box Manager for System Generator 8.x
HDL Black-Box for Synplify DSP

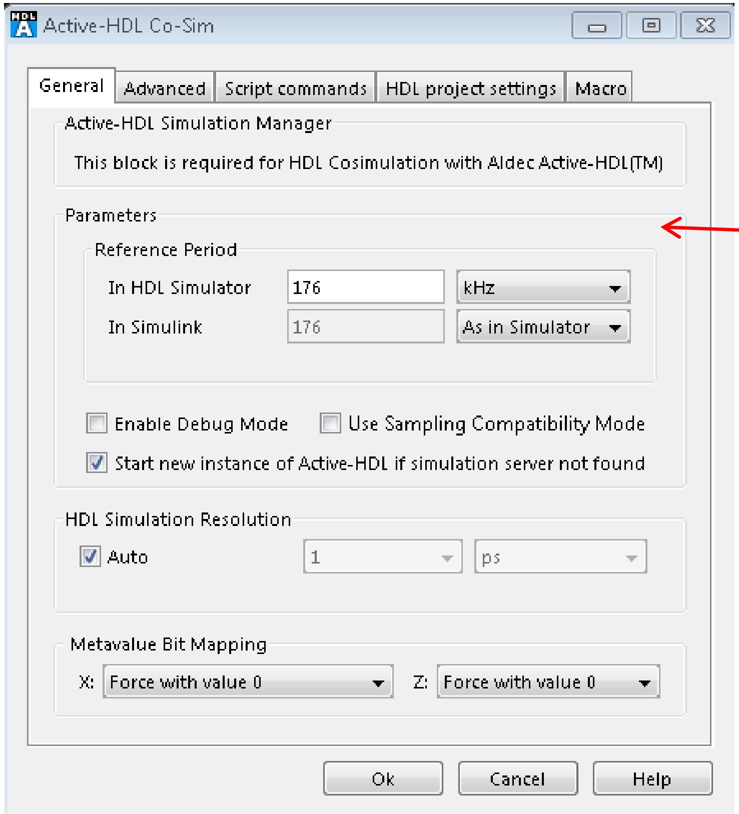
Using Active-HDL Blockset-blocks

- Following blocks are provided:
  1. Active-HDL Co-Sim
  2. HDL Black-Box
  3. HDL Black-Box Manager for System Generator 8.x
  4. HDL Black-Box for DSP Builder
  5. HDL Black-Box for Synplify DSP

  in the **Active-HDL Blockset** and other MATLAB simulation libraries to create its own simulation model.
- Create new or open the existing Simulink model and instantiate blocks included in Active-HDL Blockset. Exactly **one Active-HDL Co-Sim** block and at least one **HDL Black-Box** needs to be added to the model window in order to start co-simulation.
- The **Active-HDL Co-Sim** block is required to successfully initialize the simulation process and it can be used to define parameters of the co-simulation session.
- Each file generated by the **Generate Block Description for Simulink…** needs to be associated with the **HDL Black-Box** in order to co-simulate HDL unit described by this file.
- **HDL Black-Box Parameters** dialog box allows configuring their parameters before co-simulation.

Using Active-HDL Blockset

- To configure co-simulation parameters, double-click the Active-HDL Co-Sim symbol.



Active-HDL Simulation Manager

This block is required for HDL Cosimulation with Aldec Active-HDL(TM)

Parameters

Reference Period

In HDL Simulator: 176 kHz

In Simulink: 176 As in Simulator

□ Enable Debug Mode □ Use Sampling Compatibility Mode

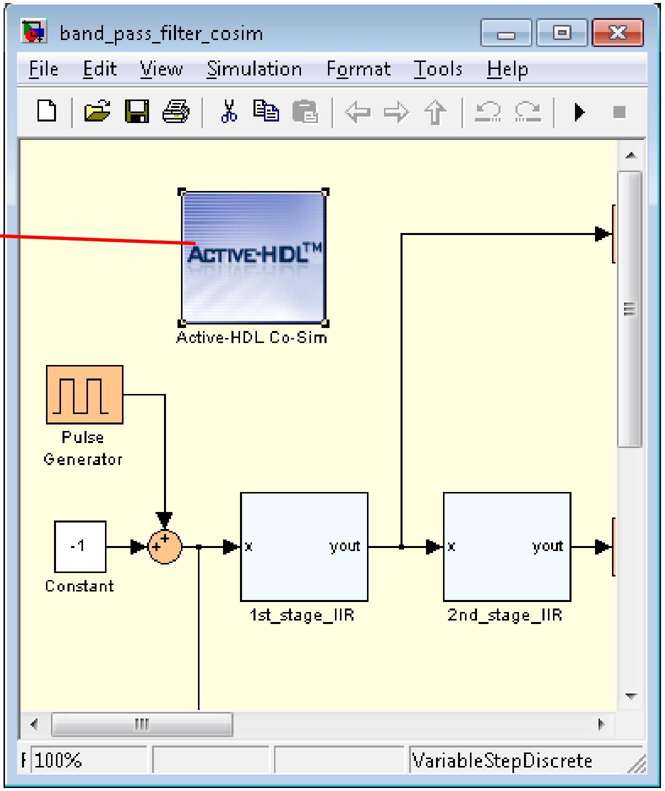
☑ Start new instance of Active-HDL if simulation server not found

HDL Simulation Resolution

☑ Auto 1 ps

Metavalue Bit Mapping

X: Force with value 0 Z: Force with value 0



Using Active-HDL Blockset

-configuring Active-HDL Co-Sim block

- **Major configuration settings available in the Active-HDL Co-Simblock:**

Reference Period

Defines relationship between Active-HDL and Simulink time domains and specifies a base sampling rate for all HDL black-boxes. The value can be specified as a fraction (1/10 or 0.1), an integer value (25), or expression (10+1/5 or 10*pi). 0

Enable Debug Mode

Allows running the co-simulation in a special diagnostic mode, where the simulation is stopped after the design has been initialized and then you can set breakpoints, watch signals, and trace HDL source code.

HDL Simulation Resolution

Allows specifying custom simulation resolution. By default this is calculated automatically based on the Reference Period.

Metavalue Mapping

Customizes the handling of unspecified values appearing on HDL Black-Box outputs.

Script commands

This tab allows to add script commands to be executed before the 'asim' command is executed and after that, as well as after co-simulation is performed. I

You can also specify the additional parameters for the 'asim' command, that will be passed to Active-HDL while initializing the simulation.

Waveform File Name

Allows specifying the name of a waveform file that will be used to display co-simulation results.

Create ASDB Simulation Database

Disables the use of the Standard Waveform Viewer and dumps simulation results to the ASDB Simulation Database that can be read by using the Accelerated Waveform Viewer.

ASDB Simulation Database Refresh Time

Specifies the refresh time for updating the ASDB Simulation Database by entering the refresh value in the ASDB Simulation Database Refresh Time (sec.) edit box. If the Use Simulation Time checkbox is marked, simulation time is used. The wall time is used otherwise. The refresh time can be defined after the Create ASDB Simulation Database option is checked.

Record Values in File

Enables generation of a file that stores test vectors for the HDL-based part of the MATLAB design. The file can be used to perform a standalone simulation with the use of Active-HDL only.

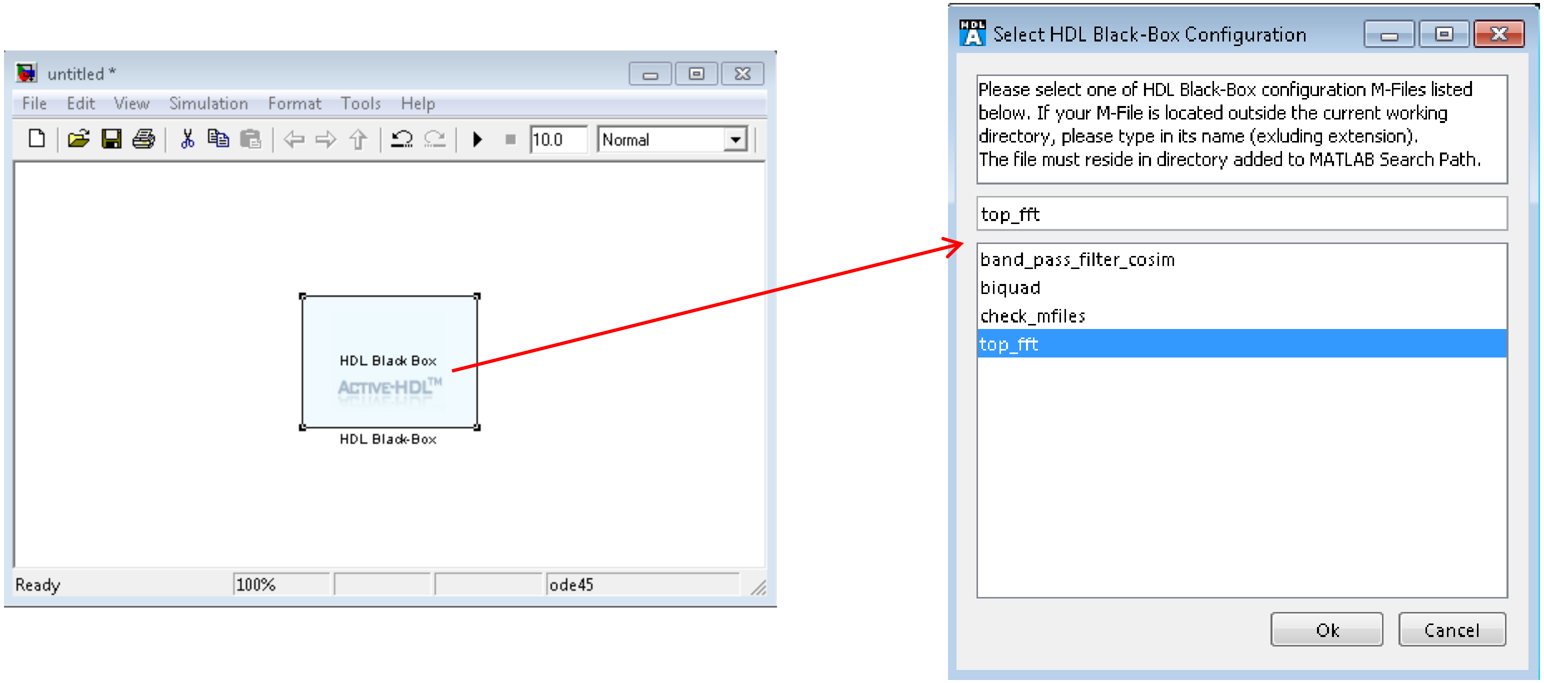
File Name

The name of the file that stores data for a standalone simulation. It can be specified after the Record Values in File option is checked.

Using Active-HDL Blockset

-selecting black box configuration

- To associate **HDL Black-Box** with an **HDL unit**, double-click the symbol and select the **block description file** generated for this unit.



Please select one of HDL Black-Box configuration M-Files listed below. If your M-File is located outside the current working directory, please type in its name (exluding extension).
The file must reside in directory added to MATLAB Search Path.

top_fft

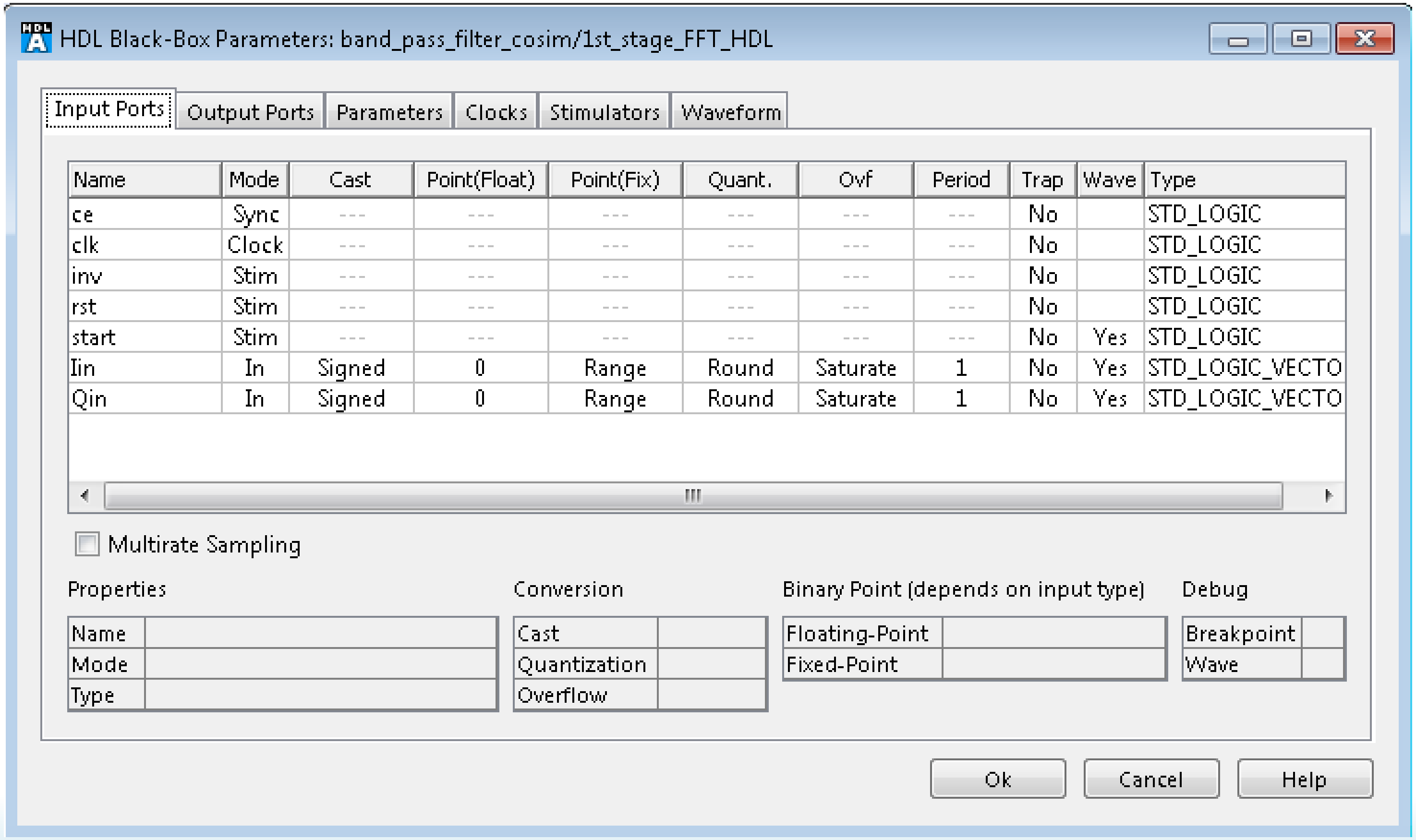
band_pass_filter_cosim
biquad
check_mfiles
top_fft

NOTE: The Select HDL Black-Box Configuration dialog box lists all configuration files placed in the current working folder. However, you can specify any configuration file visible in the MATLAB path.

Using Active-HDL Blockset

-interface signals customization

- After you selected the configuration file for HDL Black-Box, you can configure interface signals, add internal signals, specify clocks, force formulas for control signals, add signals to Active- HDL waveform etc.



Name	Mode	Cast	Point(Float)	Point(Fix)	Quant.	Ovf	Period	Trap	Wave	Type
ce	Sync	---	---	---	---	---	---	No		STD_LOGIC
clk	Clock	---	---	---	---	---	---	No		STD_LOGIC
inv	Stim	---	---	---	---	---	---	No		STD_LOGIC
rst	Stim	---	---	---	---	---	---	No		STD_LOGIC
start	Stim	---	---	---	---	---	---	No	Yes	STD_LOGIC
Iin	In	Signed	0	Range	Round	Saturate	1	No	Yes	STD_LOGIC_VECTO
Qin	In	Signed	0	Range	Round	Saturate	1	No	Yes	STD_LOGIC_VECTO

☐ Multirate Sampling

Properties

Name	
Mode	
Type	

Conversion

Cast	
Quantization	
Overflow	

Binary Point (depends on input type)

Floating-Point	
Fixed-Point	

Debug

Breakpoint	
Wave	

Using Active-HDL Blockset

-interface signals customization

- **Options located in Input Ports and Output Ports tabs control cosimulation parameters of all HDL Black-Box interface ports:**

  Name

  Displays port name of the simulated unit or name of internal signal added to black-box interface.

  Direction

  Displays the direction of port (In, Out).For ports selected in either Clocks or Stimulator tabs, displays the special function of such ports: Clock, Sync, Custom. Internal signals are indicated as Internal.

  Cast

  Specifies the numerical representation of a signal's value. Available types are: Boolean, SIGN and UNSIGN.

  Binary Point

  Specifies the binary point for a vector. Negative values and values greater than vector width are supported.

  Quantization

  Specifies the type of quantization method (available values are: truncate, round).

  Overflow

  Specifies handling method of arithmetical overflow (available values are: saturate, wrap, error).

  Period

  Specifies the sampling period of the output signal. You can specify an integer value related to Reference Period or one of signals selected as **Clock or Sync.** When you need to specify several sampling periods, check the **Multirate Sampling** option and enter required sampling periods for each output separately.

  Breakpoint

  Specifies whether a breakpoint will be set on the selected signal.

  Hidden

  Specifies whether port should be visible on the block graphical representation and thus available in the Simulink diagram

  Type

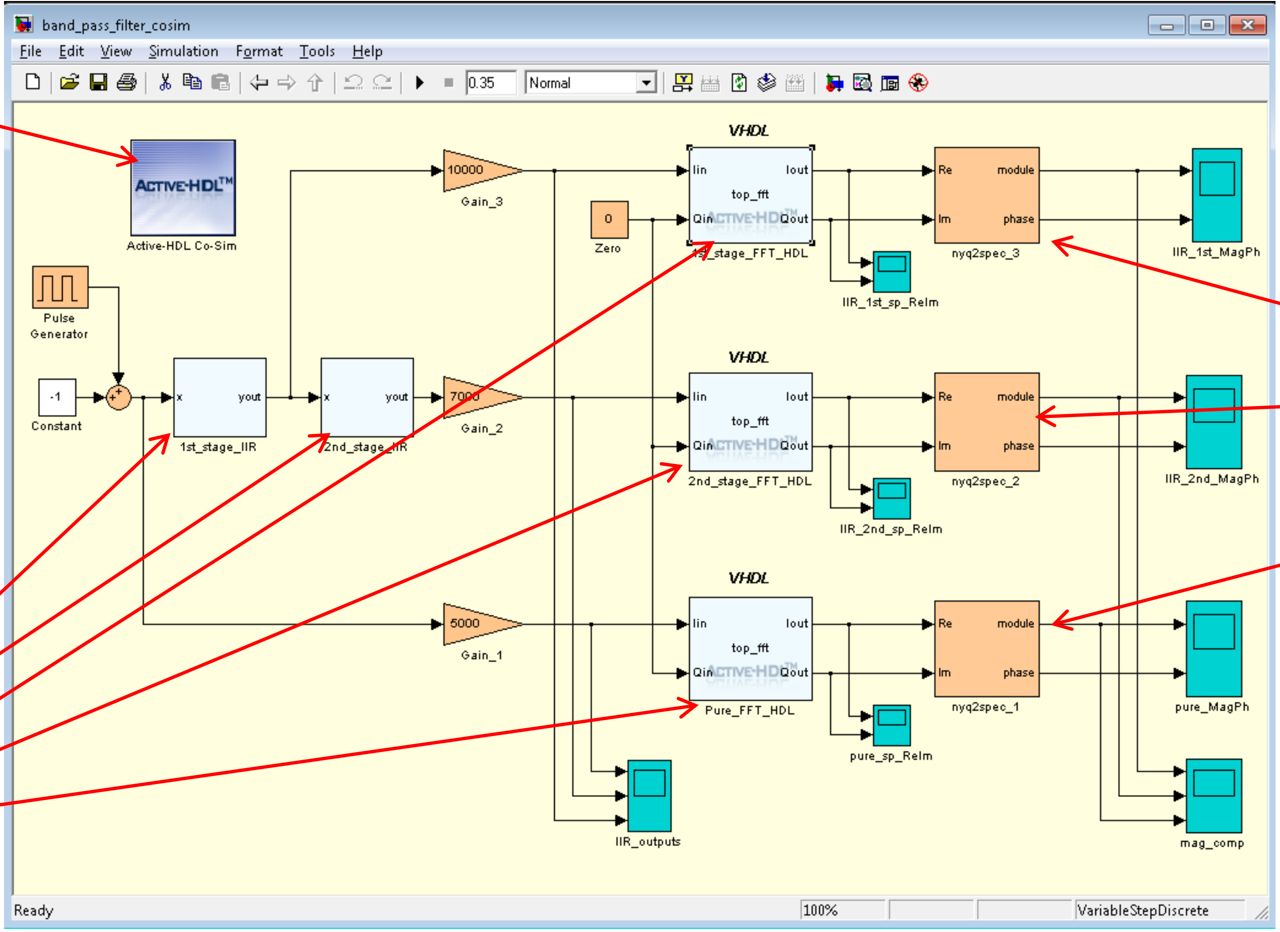
  Displays the type of the signal.

Co-Simulation

- Connection to Active-HDL

- Simulink Model with HDL Black-Box ready for co-simulation is presented below.

Active-HDL co-simulation block



HDL black boxes

Simulink blocks

The step by step example of co-simulation will be shown later in the presentation.

Using Active-HDL Blockset

-interface signals customization

1. All HDL Black-Boxes from the Active-HDL Blockset are simulated with Active-HDL simulator.
2. Running simulation from the Simulink diagram opens connection to the Active-HDL simulation server and configures it to run co-simulation of code associated with all instantiated black-boxes.
3. A new or existing instance can be configured as simulation server and is reused for all co-simulation sessions.
4. User can review results in both Simulink and Active-HDL waveform.

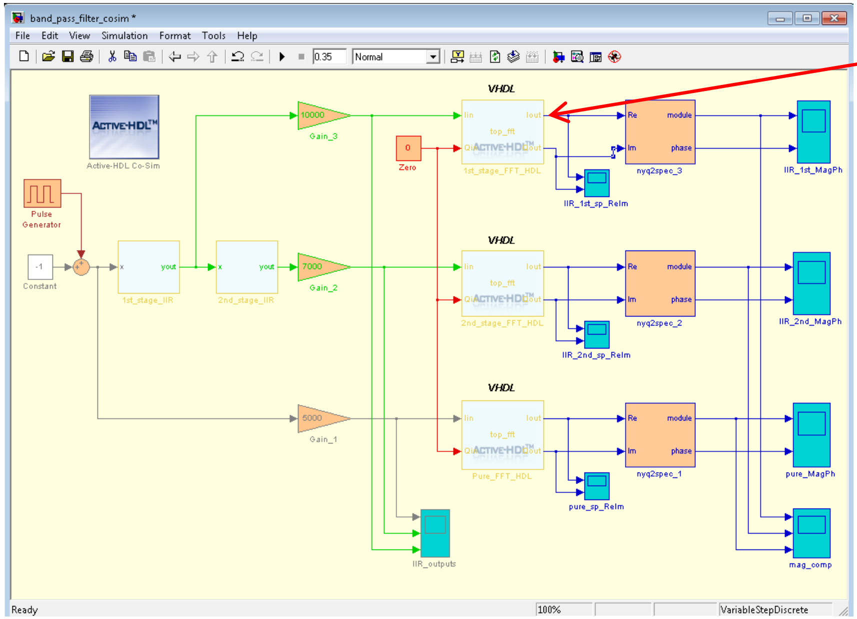
NOTE:

Waveform is created in an automated way when signals are added to list in the “**Waveform”tab.**

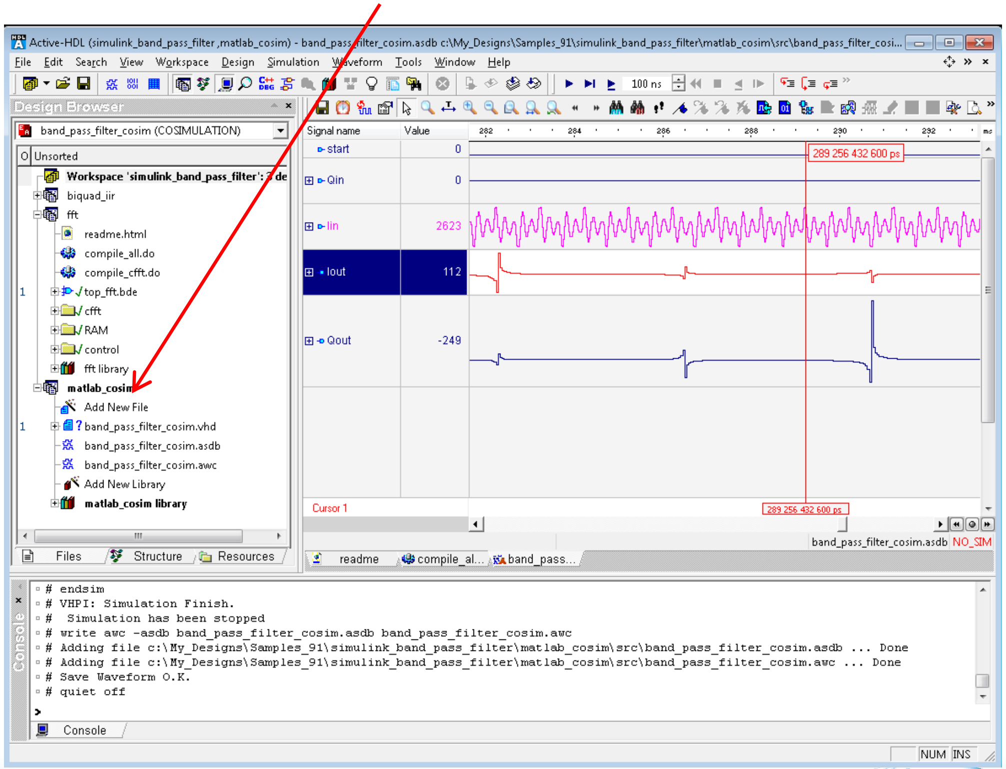
Co-Simulation

- Simulation Start

- Once Start Simulation button is pressed on the **Simulink toolbar,** the co-simulation process will be started.



HDL Black-Box simulated with Active-HDL simulator inside of the new, automatically created design.

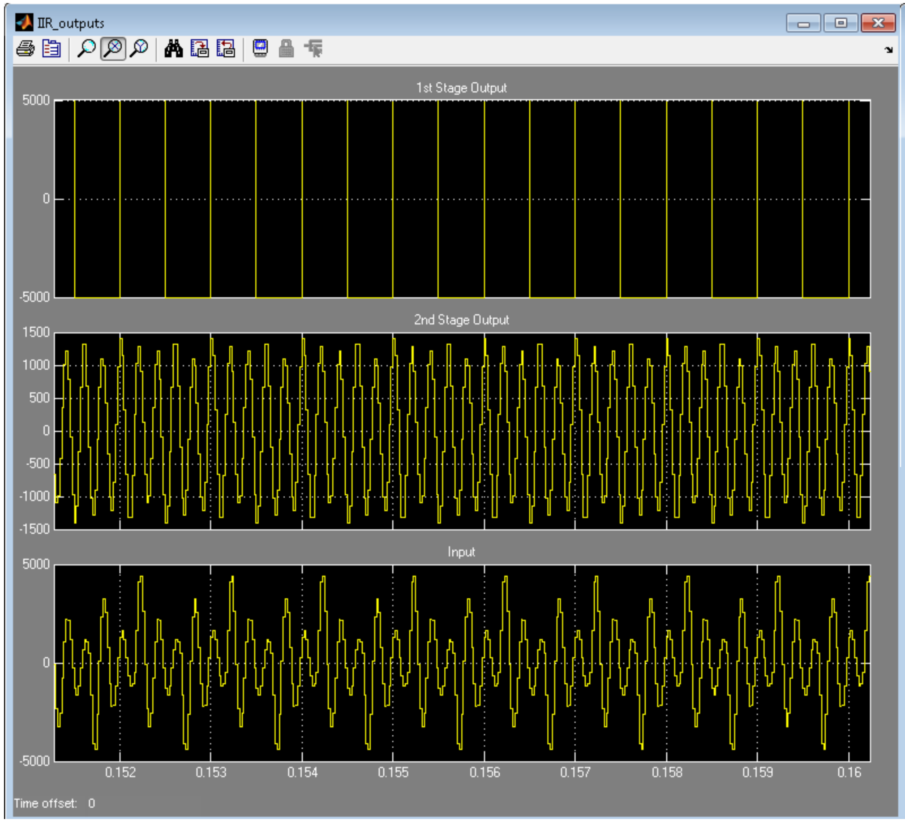

Active-HDL (simulink_band_pass_filter ,matlab_cosim) - band_pass_filter_cosim.asdb c:\My_Designs\Samples_91\simulink_band_pass_filter\matlab_cosim\src\band_pass_filter_cosi...
File Edit Search View Workspace Design Simulation Waveform Tools Window Help
100 ns
Design Browser
band_pass_filter_cosim (COSIMULATION)
Unsorted
Workspace 'simulink_band_pass_filter': 3 de
biquad_iir
fft
readme.html
compile_all.do
compile_cfft.do
top_fft.bde
cfft
RAM
control
fft library
matlab_cosim
Add New File
band_pass_filter_cosim.vhd
band_pass_filter_cosim.asdb
band_pass_filter_cosim.awc
Add New Library
matlab_cosim library
Files
Structure
Resources
Signal name
Value
start 0
Qin 0
Iin 2623
Iout 112
Qout -249
282
284
286
288
290
292
289 256 432 600 ps
Cursor 1
band_pass_filter_cosim.asdb NO_SIM
readme
compile_al...
band_pass...
Console
NUM INS

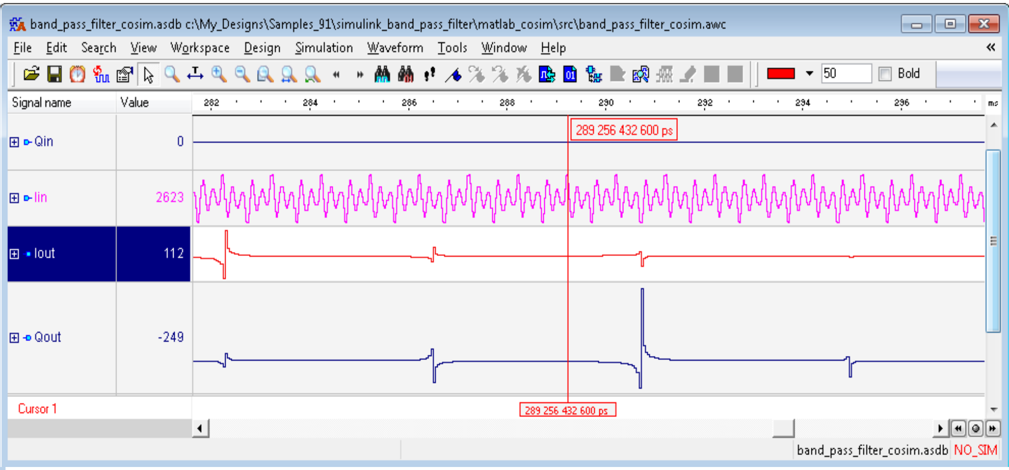
```
# endsim
# VHPI: Simulation Finish.
#  Simulation has been stopped
# write awc -asdb band_pass_filter_cosim.asdb band_pass_filter_cosim.awc
# Adding file c:\My_Designs\Samples_91\simulink_band_pass_filter\matlab_cosim\src\band_pass_filter_cosim.asdb ... Done
# Adding file c:\My_Designs\Samples_91\simulink_band_pass_filter\matlab_cosim\src\band_pass_filter_cosim.awc ... Done
# Save Waveform O.K.
# quiet off
>
```

Co-Simulation

- View Simulation Results

- User can review the simulation results within Simulink and in the Active-HDL Waveform window




band_pass_filter_cosim.asdb c:\My_Designs\Samples_91\simulink_band_pass_filter\matlab_cosim\src\band_pass_filter_cosim.awc
File Edit Search View Workspace Design Simulation Waveform Tools Window Help
50
Bold
Signal name
Value
282
284
286
288
290
292
294
296
ms
289 256 432 600 ps
Qin 0
Iin 2623
Iout 112
Qout -249
Cursor 1
289 256 432 600 ps
band_pass_filter_cosim.asdb NO_SIM

Example of use – Active-HDL design

Follow below given instructions to run example design given with Active-HDL installation.

1. Open simulink_band_pass_filter workspace from C:/My_Designs/Samples_xx folder.
2. First, use the ***compile_rtl.do*** macro in the *biquad_iir* design and ***compile_all.do*** in the *fft* design in order to compile design files and generate block description files for Simulink.
3. You can also generate block description files for Simulink manually. Select top-level modules (***biquad*** in the *biquad_iir* design and ***top_fft*** in the *fft* design) and use the **Generate Block Description for Simulink** option from the right click pop-up menu for each of the selected top-level modules. Confirm the default file names and locations by pressing OK in the **Save As** dialog box.
4. Invoke **MATLAB**.
5. If you are using the co-simulation interface for the first time, browse to the **$ALDEC/Simulink** folder and type the ***setup*** command. (The **$ALDEC** variable points to the Active-HDL installation folder; you can examine its value using the ***set*** command in Active-HDL Console).
6. Open the *$WSP/Simulink/band_pass_filter_cosim.mdl* file in the Simulink environment. Make sure that the *$WSP/Simulink* folder is either set as the current folder or added to the MATLAB search path.

Example of use – Active-HDL design

7. The model contains the Active-HDL Co-Sim block on the top-level diagram, three HDL Black-Boxes corresponding to three instances of the top_fft entity and two subsystems. Each subsystem contains one HDL Black-Box named biquad_IIR_HDL. The name displayed in the middle of the HDL Black-Box symbol corresponds to the configuration filename. By default, it is the name of the entity/module, unless you change it with the Generate Block Description for Simulink option. The HDL Black-Boxes are distinguished graphically with the Active-HDL logo.
8. Now, you can simply start simulation and observe the results on the Simulink scopes and in Active-HDL.