Course 14

Simulink® Interface in Active-HDL

Rev. Oct 11, 2012 Applies to Active-HDL 9.2 and later

Table of Contents

Introduction

The Simulink Interface built-into Active-HDL provides the integration of the Math Works' simulation tools with Aldec's HDL-based simulation environment for FPGA and ASIC designs. The interface allows designers to co-simulate functional blocks described using mathematical formulas and VHDL entities, Verilog modules, EDIF cells, as well as SystemC modules that are used as black-boxes during the verification process performed within the Simulink environment.

The interface built-into Active-HDL consists of the following components:

- Active-HDL generated block description (*.m*) file that describes an interface of an HDL black-box.
- Active-HDL Co-Sim block describing different interface properties
- Set of functions accessible from within the MATLAB Command Window.

Specifically, the Aldec tools allow you to generate *.m* (block description) file describing the interface of HDL unit (a VHDL entity, a Verilog module, EDIF cell, or a SystemC module). Once the description of the interface is generated, you can then perform co-simulation and freely customize block parameters.

Interface Specification

Software Requirements

- MATLAB R2007b or newer
- Active-HDL 9.1 or newer

Language Support

- VHDL IEEE Std 1076-1993
- Verilog IEEE Std 1364-1995
- EDIF 2 0 0

Supported VHDL Data Types

- Boolean
- Signed, Unsigned (up to 256 bits for vectors), Integer, Real
- BIT and BIT_VECTOR
- STD_LOGIC, STD_ULOGIC, STD_LOGIC_VECTOR, STD_ULOGIC_VECTOR

Supported Verilog Types

- Logic scalars and vectors up to 256 bits (wire, reg)

HDL Block Port Types

- Regular HDL Black-Box: DOUBLE, SINGLE, Int8, uInt8, Int16, uInt16, Int32, uInt32, Fix and uFix up to 128 bits

Conversion Options

- Port Cast for inputs: Signed (2's comp), Unsigned, Boolean
- Port Cast for outputs: Signed (2's comp), Unsigned, Boolean, Int8, uInt8, Int16, uInt16, Int32, uInt32, Fix, uFix
- Vector Width: Integer values between 1 and 256, the use of generics is allowed
- Position of binary point (fractional part): unlimited
- Quantization (HDL Black-Box inputs only): Truncate, Round

- Overflow (HDL Black-Box inputs only): Saturate, Wrap, Error

Interface Setup

Run Active-HDL Setup Program

Follow procedures described below for setting up the Simulink Co-Sim interface:

1. Start MATLAB.
2. Change the Current Directory in the MATLAB window to the $ALDEC\Simulink directory.
3. Enter the *setup* command in the MATLAB Command Window and press Enter. The appearing warning dialog box prevents you from accidental removing previous versions of the Active-HDL Blockset installed in MATLAB. Press ‘*Yes*’ to continue. The following message should be displayed in the Command Window:

```
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.
```

4. After the setup is finished, you are able to use Active-HDL Blockset immediately. If the Simulink Library Browser or Help window is already open, you may need to restart MATLAB in order for the changes to take effect.
5. When you run the **Simulink Library Browser** by typing *simulink* in **Command** Window or clicking on the **Simulink** toolbar button, you will see the Active-HDL Blockset in the left pane. The blockset contains the following blocks:
    - Active-HDL Co-Sim
    - HDL Black-Box
    - HDL Black-Box Manager for System Generator 8.x
    - HDL Black-Box for Synplify DSP

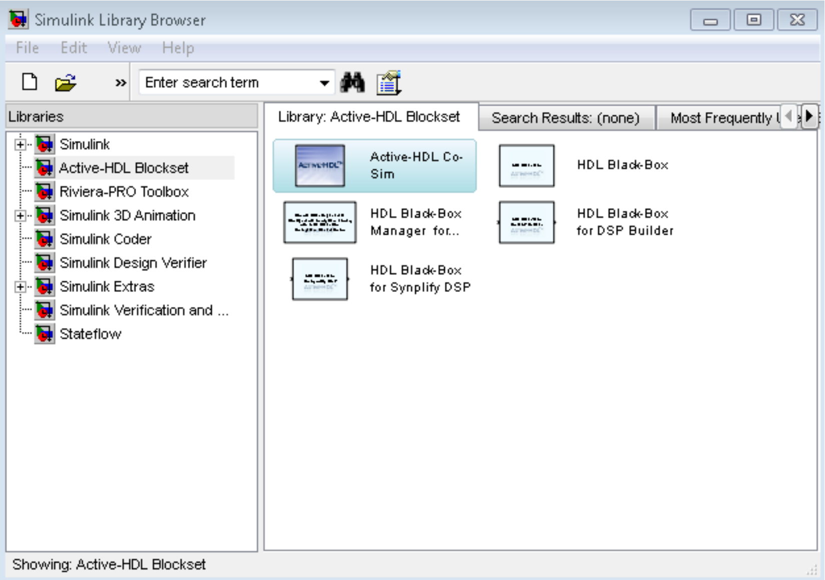


Figure 1 Simulink Library Browser

Note: If you run into problems running setup due to rare platform-specific issues, you will need to set up the interface manually. Please refer to **Simulink® Interface** topic inside Help Documentation for detailed instructions.

FFT Program Example

Scope of this example

This example educates you about the interface functionality of Active-HDL and Simulink. It assumes that you know basic functionality of Active-HDL and Matlab-Simulink.

Introduction

This example is based upon the simulink_band_pass_filter design which comes with Active-HDL installation. This design demonstrates the usage of Simulink interface. simulink_band_pass_filter is a simple DSP design using IIR bandpass filter and 1024-point FFT components.

The purpose of this design is to demonstrate the use of Active-HDL Co-Simulation Interface to Simulink® in case of different sampling rates required for different HDL blocks. It shows how to synchronize Simulink sampling with HDL simulator and how to differentiate sampling rates of HDL blocks instantiated in Simulink diagram.

The top-level block diagram is created using Simulink block diagram editor. This diagram instantiates black-box of the FFT and IIR filters. The FFT design is described in VHDL while the IIR filters are described in Verilog.

The workspace simulink_band_pass_filter contains a description of components used in Simulink block diagram. It includes:

- biquad_iir component containing description of the second order IIR filter.
- fft component containing descriptions of the Complex FFT module, RAM memory to store FFT data and control logic responsible for proper FFT output data order.

The Simulink design has a testbench for two IIR filters and three FFT blocks. There are two band-pass filters. Each of them contains single biquad band-pass section. Three FFT blocks process input, first-stage output and second-stage output signals of bandpass filter.

The filter is stimulated with 1 kHz square signal. Two identical IIR filters are designed as Chebyshev bandpass filters with a 44 kHz sampling frequency and corner frequencies of 4.5 kHz and 5.5 kHz, so they pass the 5th harmonic component of source square signal. To have the 5th harmonic in the middle of FFT output spectrum, the whole bandwidth is set to 11 kHz, which gives the FFT sampling frequency of 22 kHz.

Generating Block Description File for Simulink Interface

Before an HDL unit can be co-simulated, interface descriptions file (M-file) for that unit needs to be generated. The file describing an interface (*.m*) of a VHDL entity/architecture pair, a Verilog module and EDIF cell can be generated by using the Active-HDL GUI or command line.

Generate the M-file in Active-HDL GUI

1. Open a workspace and set an active design that contains units to be co-simulated.

   Note: Using workspace and designs is not obligatory but it is strongly recommended as it provides better design management.

2. Compile source files that describe the design unit that will be used as a black-box in Simulink.
3. Expand the files in design browser and select the **Generate Block for Simulink** option from the context menu as shown in the figure below.

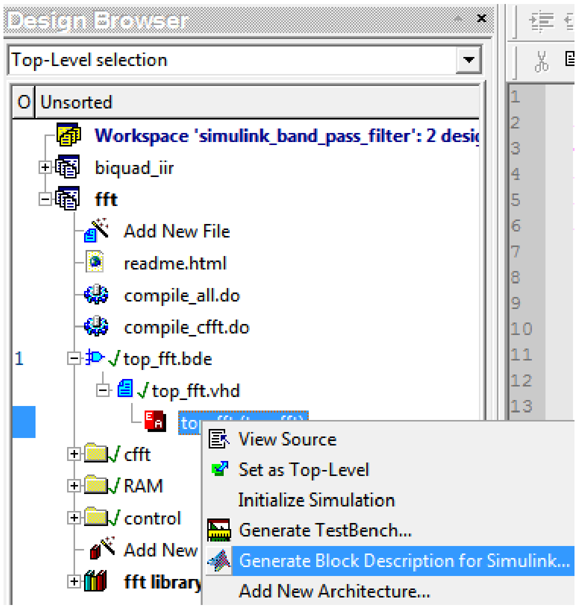


Figure 2 Generating block description for Simulink

4. The **Generate Block for Simulink** dialog box opens. Verify the location of the generated blockset description file and press **Save** to close the window.

5. Alternatively, you can also generate the M-file from the command line use the `simulinkgenmod` command in the Console window.

   Example

```
simulinkgenmod -overwrite -o $wsp/../Simulink/top_fft.m top_fft
```

NOTE: Before you move to next step you have to repeat same steps to generate M-file for another component (biquad_iir).

Using Active-HDL Blockset

Before you can run co-simulation in Simulink, you need to instantiate blocks available in the Active-HDL Toolbox in a Simulink model window. You may also want to modify their parameters.

Instantiating Active-HDL Blockset blocks in Simulink Diagram

- Switch to MATLAB. Change the current directory and set it to the location of the simulink/ directory storing the library of black-boxes created in Active-HDL (in the Generate Block for Simulink dialog box).
- Click the Simulink icon from MATLAB main toolbar or type simulink in the MATLAB Command Window to open the Simulink Library Browser window.
- In the left pane of the Simulink Library Browser window, the Active-HDL Blockset is displayed. The contents of the blockset are displayed in the right pane. This version of the blockset contains the following blocks:
  a. Active-HDL Co-Sim
  b. HDL Black-Box
  c. HDL Black-Box Manager for System Generator 8.x
  d. HDL Black-Box for Synplify DSP
- To add the Active-HDL Co-Sim block to the Model window use the Add to the current model command or drag-and-drop.

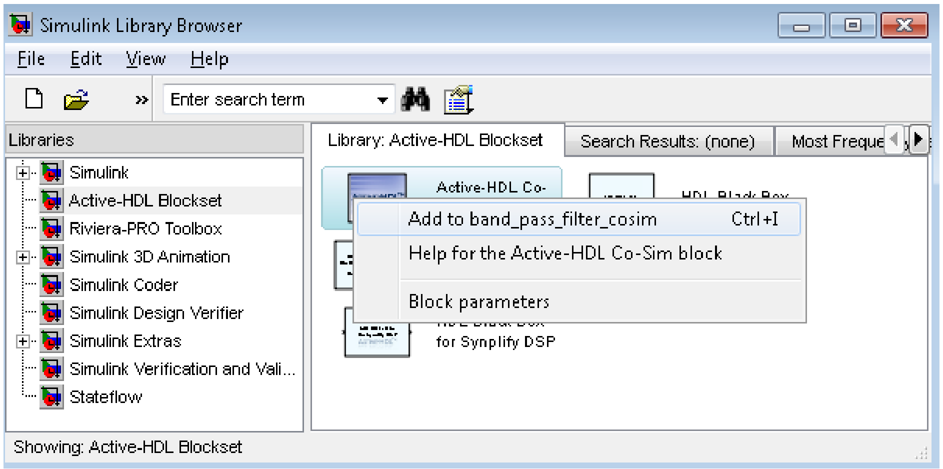


Figure 3 Simulink Library Browser

- Select HDL Black-Box and add it to the same Simulink model window. A Select HDL Black-Box Configuration dialog box should pop-up after you dragged-and-dropped the block template from the Simulink Library Browser window into the model window. There, please specify the functionality and associate an HDL model which was generated before (block description file- .m) by selecting the black-box configuration file.
- By Using HDL-Black-Box Parameter window you can control some of the Active-HDL features for waveform, stimulus and breakpoint etc.

Configure Timing Dependencies between Active-HDL and Simulink

One of the most important issues for co-simulation between two simulators is the time-domain synchronization of the two simulation runs. Following aspects must be properly considered in order to successfully configure the co-simulation:

- relation of time-domains between simulators
- relations of sampling period between ports of all HDL Black-Boxes instantiated in the Simulink model
- configuration of clock ports of HDL Black-Boxes

The Active-HDL Toolbox provides all features required to flexibly handle the synchronization of the simulators and time parameters of the ports of the HDL Black-Boxes in both simulators.

Those timing related dependencies are set either in the Active-HDL Co-Sim block or HDL Black-Box parameter dialog box. You can set the following important timing related dependencies:

- Configuring the Relation of Time-domains between Simulators
- Selecting the Reference Period
- Configuring Sampling Period of the HDL Black-Box Ports
- Configuring Clock Ports of HDL Black-Boxes
- Using Optional Synchronization Signals

Active-HDL Co-Sim Dialog Box

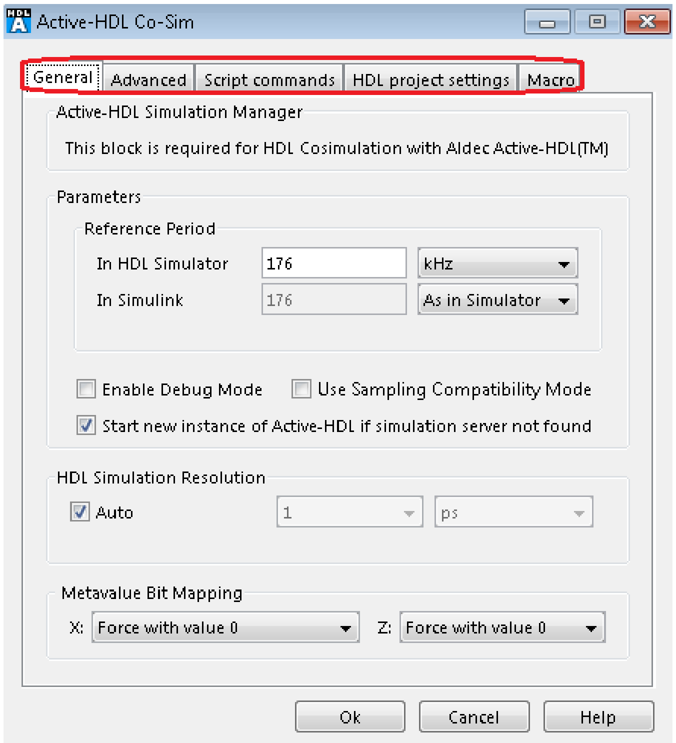
1. Double-click the **Active-HDL Co-Sim block** to open the Active-HDL Co-Sim dialog box. It allows you to configure the sampling period and the relationship between Active-HDL and Simulink time domains as well as general co-simulation settings.
2. Specify new settings and apply the changes clicking the OK button. This dialog box provides different settings necessary for pre-initialization and post-initialization as well.

You may use the following tabs of Co-Sim dialog box to configure the co-simulation parameters:

- **General tab:**
    - **Parameter Reference Period** defines relationship between Active-HDL and Simulink time domains and specifies a base sampling rate for all HDL black-boxes. In the figure Reference period for HDL simulator is given 176 kHz frequency so in case of frequency unit resulting sampling period is calculated as reciprocal of entered value.
    - You can set **Enable Debug Mode** which will run co-simulation in a special diagnostic mode. In this mode, the simulation is stopped after the design has been initialized, which then allows you to set breakpoints, watch selected signals, and trace HDL source code inside Active-HDL.
    - You can set **Start simulation in console mode**. You can set HDL simulation resolution.
    - You can decide mapping of **Metavalue**.

  As mentioned before we have two different entities having two different frequencies, so we want to simulate blocks with two different sampling frequencies. The FFT runs with 22 kHz, which gives the period of 45.45us, while IIR runs with 44 kHz, which gives the period of 22.73us, twice shorter than FFT period. The resulting maximum clock period is 22.73us and, consequently, possible reference periods are: 11.36us, 7.58us 5.68us, 4.55us, 3.79us, 3, 25us 2.84us and so on. The corresponding frequencies of 88 kHz, 132 kHz, 176 kHz, 220 kHz, 264 kHz, 308 kHz, 352 kHz, etc. also can be used in the Reference Period section.

NOTE: Be aware of the round-off error when specifying a period by using frequency units. The fractional part of the resulting period will be rounded to three digits. All parameters specifying the time are defined as integers related to the Reference Period.



Active-HDL Co-Sim

General | Advanced | Script commands | HDL project settings | Macro

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

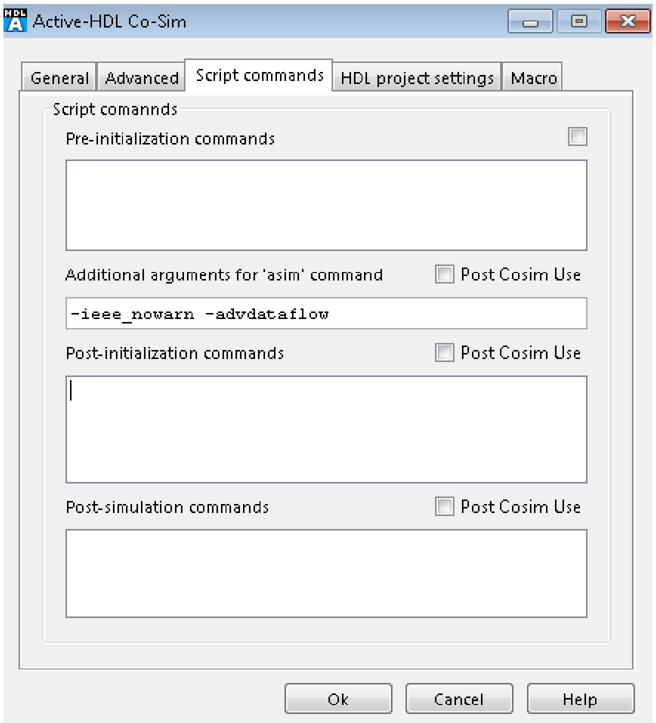
☑ Auto 1 ps

Metavalue Bit Mapping

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

Figure 4 Active-HDL Co-Sim Dialog Box (General Tab)

- **Advanced tab:** It allows you to give waveform file name in which co-simulation results will be displayed. It also allows you to save test vectors for HDL part of you Simulink design for both driving values as well as response value so later you can use this file for standalone simulation without use of the Simulink.
- **Script Commands tab:** it allows you to set `pre / post -initialization` commands and post simulation commands. You can set different arguments (switches) for *asim* command also. In figure below we have mentioned some *`asim`* switches like *`no ieee warning`* and enable advanced dataflow. You can set Post-initialization commands like enabling code different coverage etc. You can also have different waveform related commands in Post-simulation commands window.



Script comannds

Pre-initialization commands □

Additional arguments for 'asim' command □ Post Cosim Use

```
-ieee_nowarn -advdataflow
```

Post-initialization commands □ Post Cosim Use

Post-simulation commands □ Post Cosim Use

Figure 5 Active-HDL Co-Sim Dialog Box (Script commands)

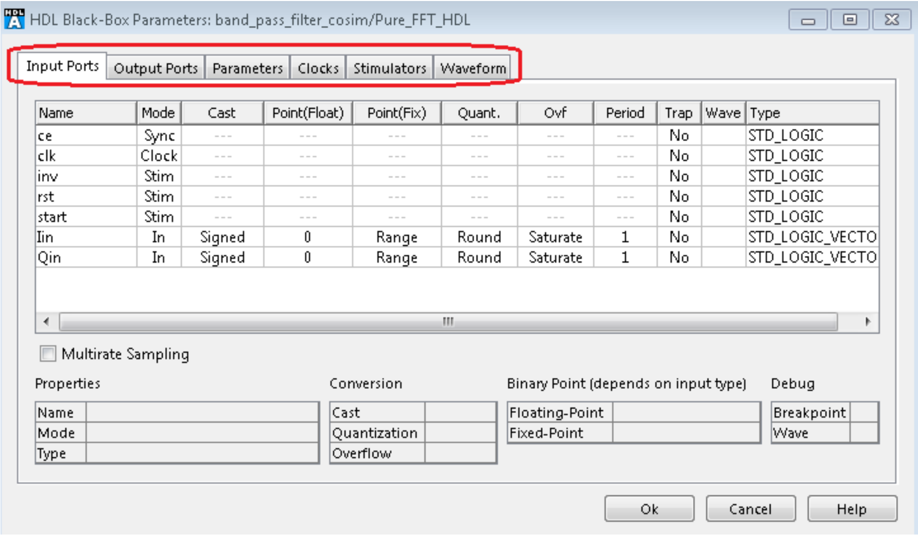
- **HDL Project Settings tab:** You can set '*use following HDL workspace / files*' for co-simulation. Simulator path can be set here as well.
- **Macro file tab:**
  - Use **macro enables** specifying a macro file to be executed in Active-HDL before the co-simulation run. Here you can set different compilation options you want to have in your design compilation like debug, incremental options (switches) etc.
  - **Use macro typed below** enables specifying in the text box below a set of commands to be executed in Active-HDL during compilation time.

Specifying HDL Black-Box Parameters

- Double-click on a Black-Box (VHDL model) symbol for what you want to edit the configuration. The **HDL Black-Box Parameters** dialog box opens. The dialog box provides a set of block-specific options (depended on block type) that have been grouped in the following categories:

- **Input and Output Ports:** Lists all objects that will be simulated and/or observed during co-simulation.
  - Mode: Shows the direction of data transfer or a mode for special signals. The Stim mode indicates that a signal has been added to the Stimulators tab, Clock and Sync indicate that signals have been added to the Clocks tab.
  - Period: This column is important. It defines the update rate for input and output. As shown in figure below sampling period is set to 1 this means that data between Simulink and HDL blocks will be exchanged in time intervals equal to the Reference Period (set in Co-Sim block – 176 kHz). For an example if Period is set to 2 then rate of update will be reference period divided by 2 (88 KHz).
  - Point: Allows specifying the binary point for a vector. As HDL Black-Box accepts input data of both floating- and fixed-point types, two separate options are provided:

  - Point (Float) - double numbers in Simulink are converted to binary numbers in fractional two's complement notation. This parameter defines position of the 2^0 bit.
  - Point (Fix) - the point value can be inherited from an input fixed-point number (by setting Range or Accuracy) or customized independently (value set for float-point types can be used by setting As for Float value).
- **Parameters:** Displays port generics/parameters defined in a black-box interface; if necessary, an actual value of a generic/parameter can be modified. See Parameters for details.



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		STD_LOGIC
Iin	In	Signed	0	Range	Round	Saturate	1	No		STD_LOGIC_VECTO
Qin	In	Signed	0	Range	Round	Saturate	1	No		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

Figure 6 HDL Black-Box Parameters (Input Ports Tab)

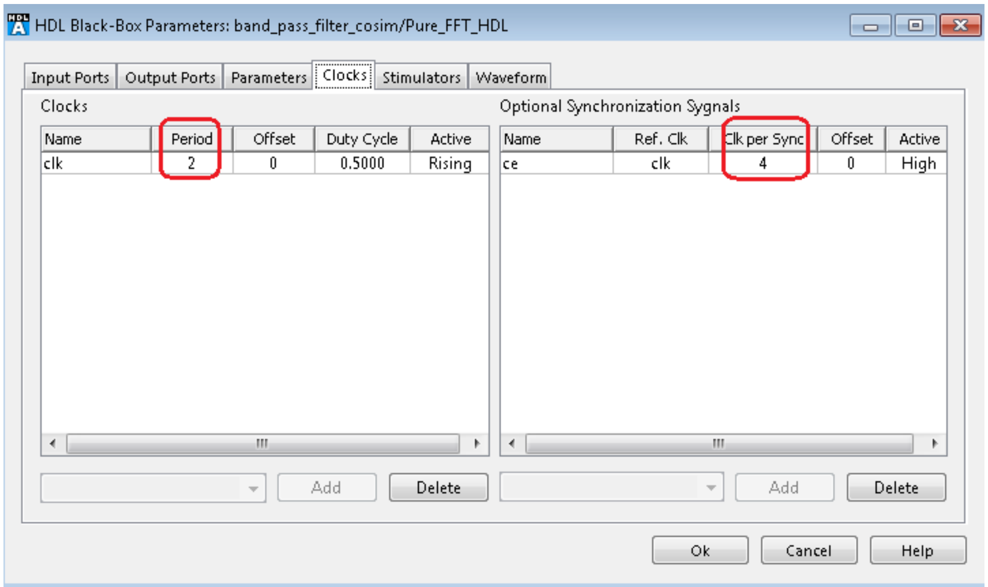
- **Clocks:** (as shown in Fig 7) allows precise specification of model synchronization signals such as Clock, Clock Enable, or other synchronization signals.

As mentioned above in the **'General'** tab of Active-HDL **'Co-Sim Dialog Box'** section that there are two different operating frequency entities (FFT block – 22 kHz & IIR – 44 kHz). These differentiating the sampling period can be accomplished in two ways: by using one clock frequency for all HDL Black-Boxes and additional synchronization signals to activate block, or by using multiple clock frequencies directly.

Clock period here is product of reference period and integer value given in **'Period'** column. In optional synchronization signals we have enable (*'ce'*) signal which can be used to activate one HDL black box slower than the other HDL black box.

For example Figure 7 shows the HDL Black-Box Parameters of one of the FFT blocks:

- **Settings for the clk port selected as the Clock signal:**
- **Period:** 2 - forces clk period of 11.36us (it corresponds to 88 kHz, It divides reference period by integer number given here)
- **Settings for the ce port selected as the Sync signal:**
- **Reference Clock:** clk - uses clk Clock as reference
- **Clk per Sync:** 4 - activates the ce port during every 4th clk period, which results in the block activation period of 45.45us (it corresponds to 22 kHz)



Clocks

Name	Period	Offset	Duty Cycle	Active
clk	2	0	0.5000	Rising

Optional Synchronization Sygnals

Name	Ref. Clk	Clk per Sync	Offset	Active
ce	clk	4	0	High

Figure 7 HDL Black-Box Parameters (Clocks Tab)

- **Stimulators:** Allows entering user-defined formulas (in the format compatible with the *'force'* command in HDL simulators) and defining stimuli for inputs that have not been selected in the Clocks tab.
- **Port Settings:** It allows you to select ports you want to be in waveform by clicking inside Wave column and can customize signal properties (e.g. radix, notation) for each port. Here you can set signal breakpoints too by double clicking in **'Break column'**.

- Once you are done specifying HDL Black-Box Parameters click **Ok.**

Note: You have to follow same steps to customize HDL black-box parameters in another component (biquad_iir) before you proceed to next step.

Running Co-simulation

- Now you can complete Simulink diagram by connecting different HDL Black-Boxes to other Simulink components as per you design requirements as shown in **Figure 8**.

Preparing Active-HDL for co-simulation

- In order to set Active-HDL as an HDL simulation server, it needs to be switched to the simulation server mode. By default, Active-HDL is not working as the simulation server. There are two approaches of setting up Active-HDL as the simulation server:
- One of existing instances of Active-HDL is switched to the simulation server mode and used for all co-simulation sessions.
- A new instance of Active-HDL is started automatically during the first simulation and reused for subsequent sessions.

If an instance of Active-HDL is already running, you can switch it to the server mode by issuing the following command in the Console window:

```
remote_on
```

After the simulation server initializes properly, you will see the following message:

```
SERVER: Active-HDL simulation server is ready.
This instance of Active-HDL will execute commands send by external process.
In order to stop simulation server, use the remote_off command.
```

Starting co-simulation

- In order to start simulating your model, choose the Start command from the Simulation menu of the Simulink model window or click the Start simulation toolbar button. When you initiate co-simulation in the Simulink environment, a new design matlab_cosim is created and added to the workspace. Active-HDL loads the new workspace. Previously, the workspace contained designs that were used to generate block description files (*.m*).

The $wsp/*band_pass_filter_cosim* design contains **automatically generated wrapper** that binds all HDL components/modules required to co-simulate HDL black-boxes instantiated in the Simulink model that is shown in Figure 8. In the figure 8 boxes in red are VHDL-models (black boxes for Simulink) generated in Active-HDL and box in blue is interface module which handles all the parameter details regarding interface. Remaining modules are native Simulink modules.

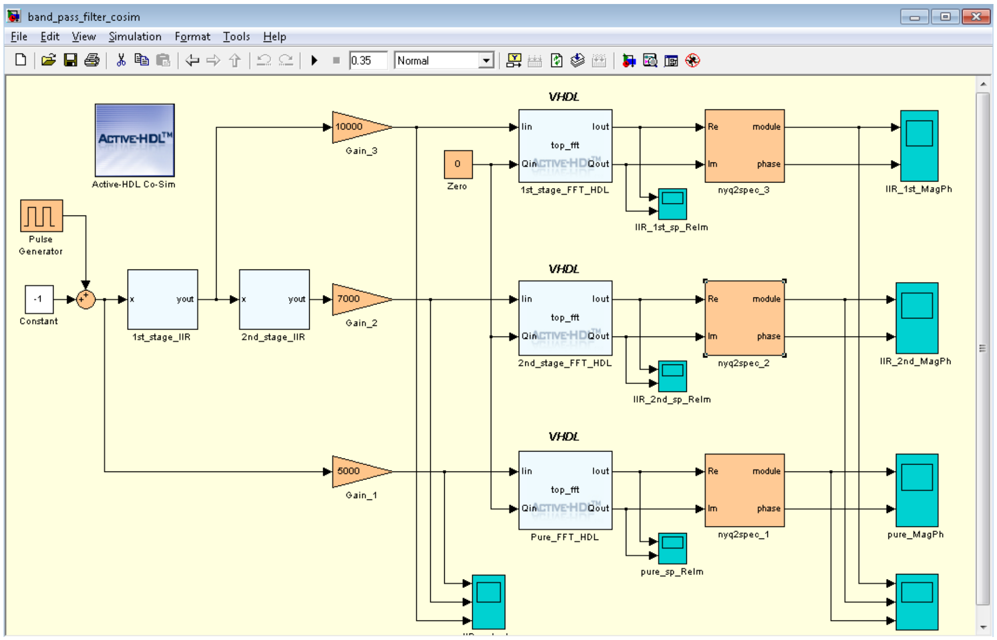


Figure 8 example model window

When the workspace is generated and the co-simulation has been initialized, the design is compiled and, in the next step, the HDL simulation session is run. The entire process is executed and based on commands passed by Simulink to Active-HDL through the co-simulation interface. The commands execution can be monitored in the Console window.

Simulation Results

Once simulation is ended Active-HDL waveform window gets populated with different wave form.

You can convert waveform display of signals to analog display and compare these waveforms with the Simulink generated waveforms and analyze your results. To convert signals in analog format, select the signal in waveform and right click **Properties | Display tab** and select analog.