Running Simulation & Accelerated Waveform Viewer

Learn how to run simulation and use waveform viewer

Simulation

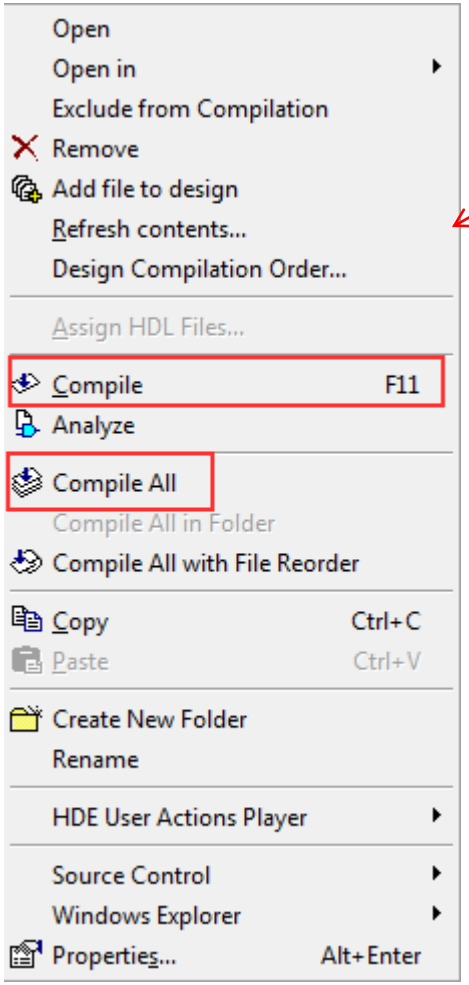
- **Simulation steps:**
  - Compile the design
  - Set the top-level architecture
  - Open Waveform Editor
  - Drag the signals
  - Initialize simulation
  - Apply stimulators
  - Advance simulation
  - Verify results
  - Save simulation run

Compiling Design

- Open the freq_meter sample design, accessible in VHDL_Designs category of the **Workspace/Design Explorer.**
- Before you start simulation, you must compile the design files to reflect the latest changes. Remember that saving the source file is not enough. Simulation is based on the entities compiled into the working library.
- To compile the design you can choose the **Compile All** option from the pop-up menu in the **Design Browser** window. You can also call the same commands from the **Design** menu.

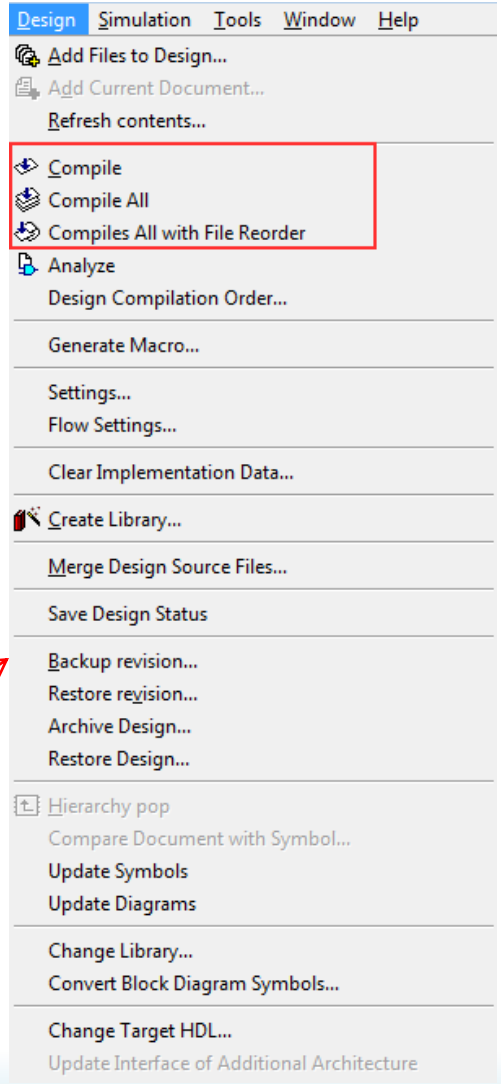
Note: The Compile All with File Reorder command automatically compiles all design files in the specified compilation order.

Compiling Design



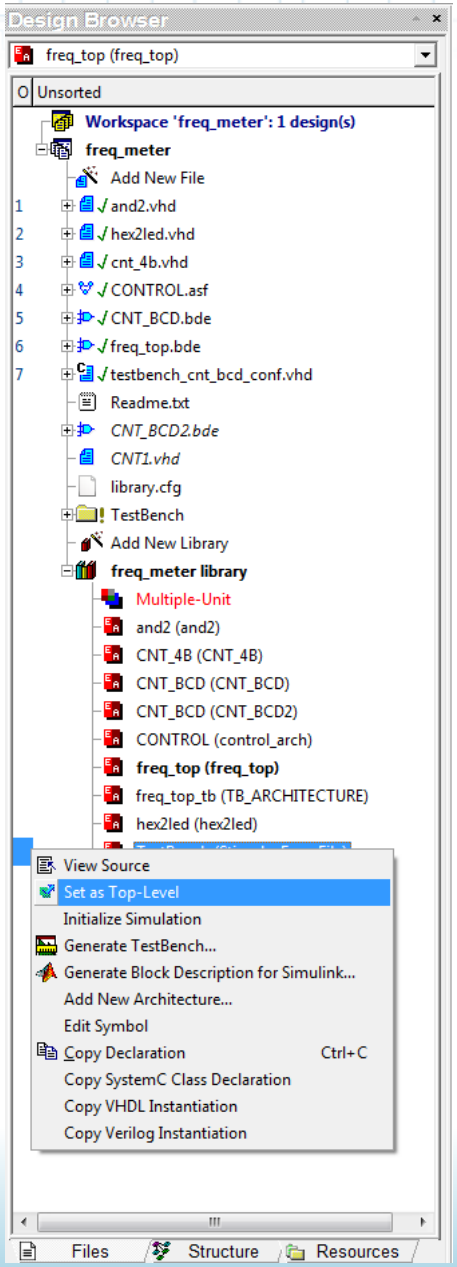
Right Click context menu from **Design Browser**

Design menu from main toolbar



Setting top-level

- Simulation is carried out for the selected library architecture called top-level architecture.
  - To choose a top-level architecture, expand the working library in the Design Browser window.
  - Select the testbench (stimulus from file) entity/architecture pair and choose the **Set as Top-Level** option from the pop-up menu.

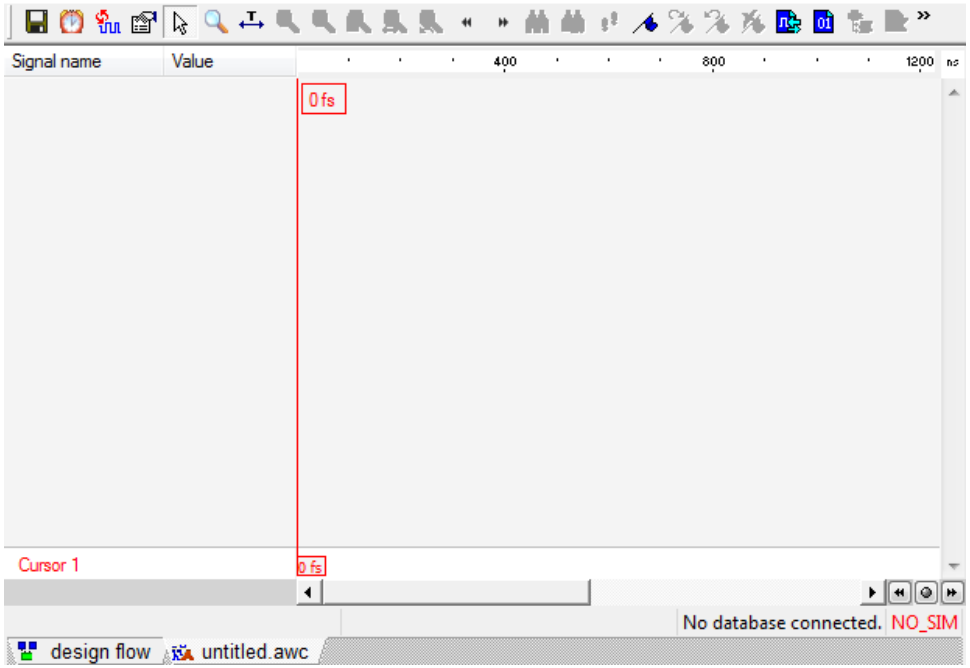


Opening Waveform Editor

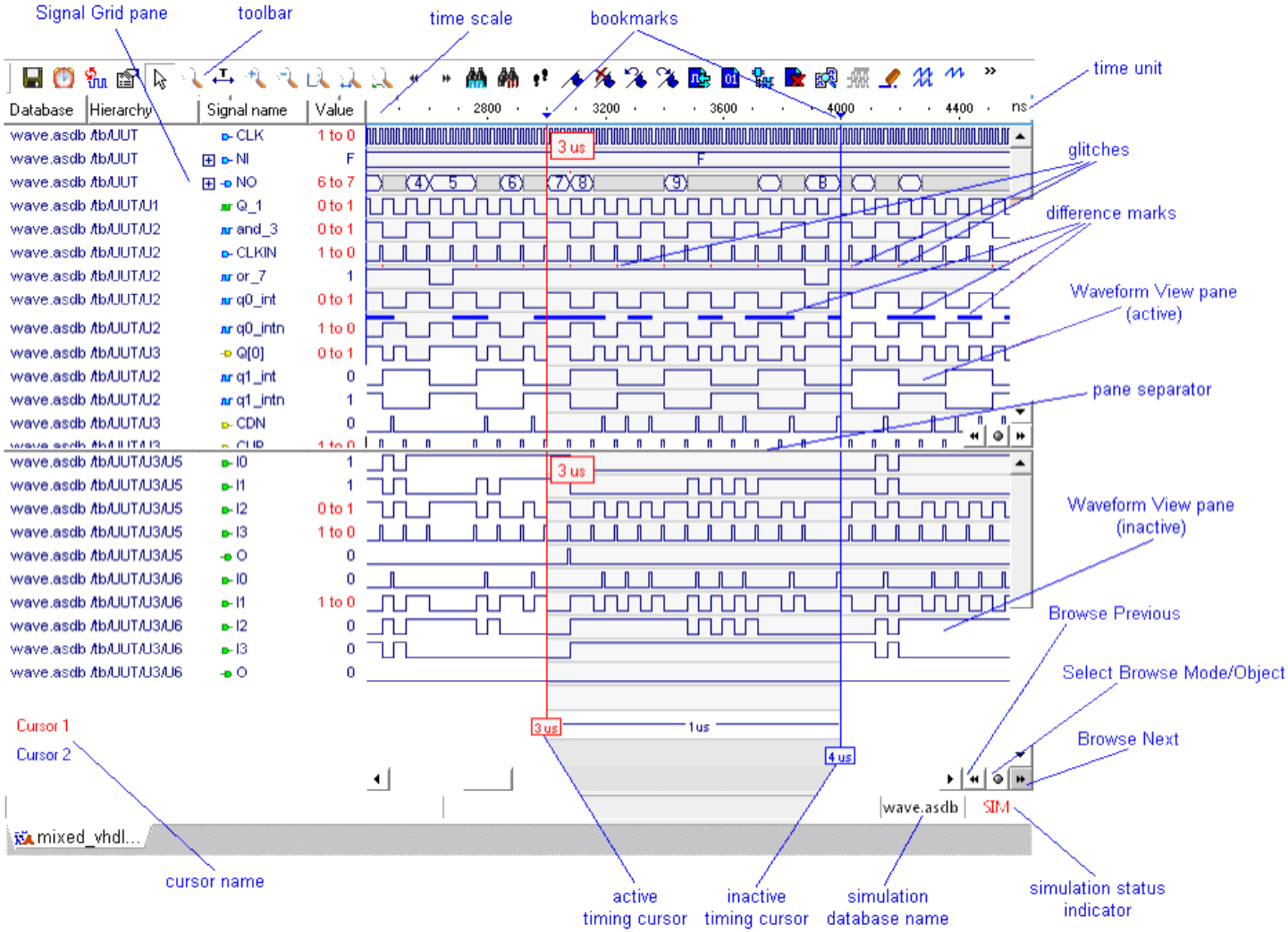
- Simulation results are displayed in the **Waveform Editor** that is a tool designed to observe simulation results as timing waveforms and to edit simulation test vectors.

  To open the **Waveform Editor**:

  - Click the toolbar button
  - Choose **New | Waveform** option from the **File** menu.

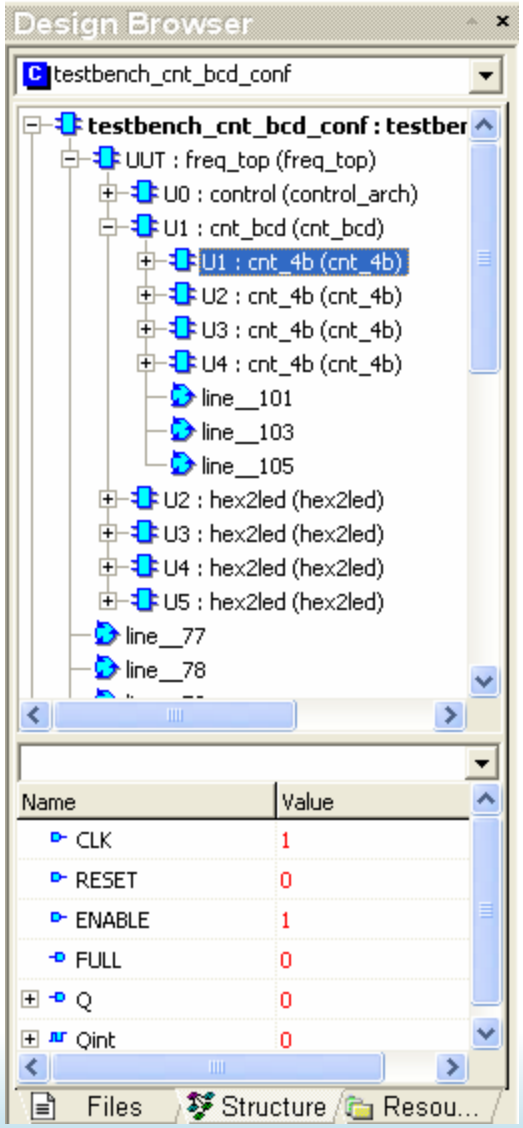


Waveform Editor



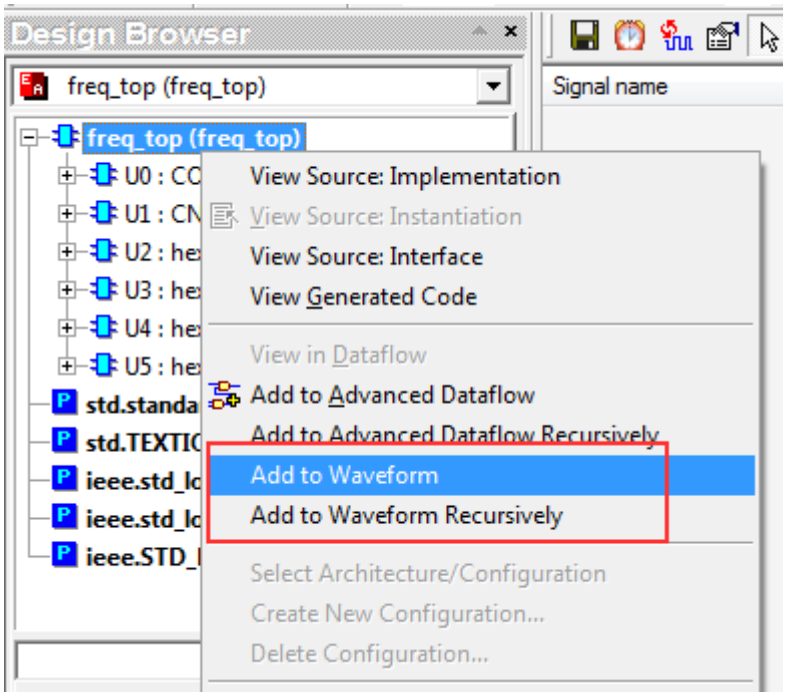
Adding Signals

- To see the results of simulation, you have to add the signals to the Waveform Editor window. You can either drag them from the **Design Browser** window or use the **Add Signals** window.
  - In the **Design Browser** window, switch to the **Structure** tab and select a particular component.
  - Select the signals and drag them to the **Waveform Editor** window



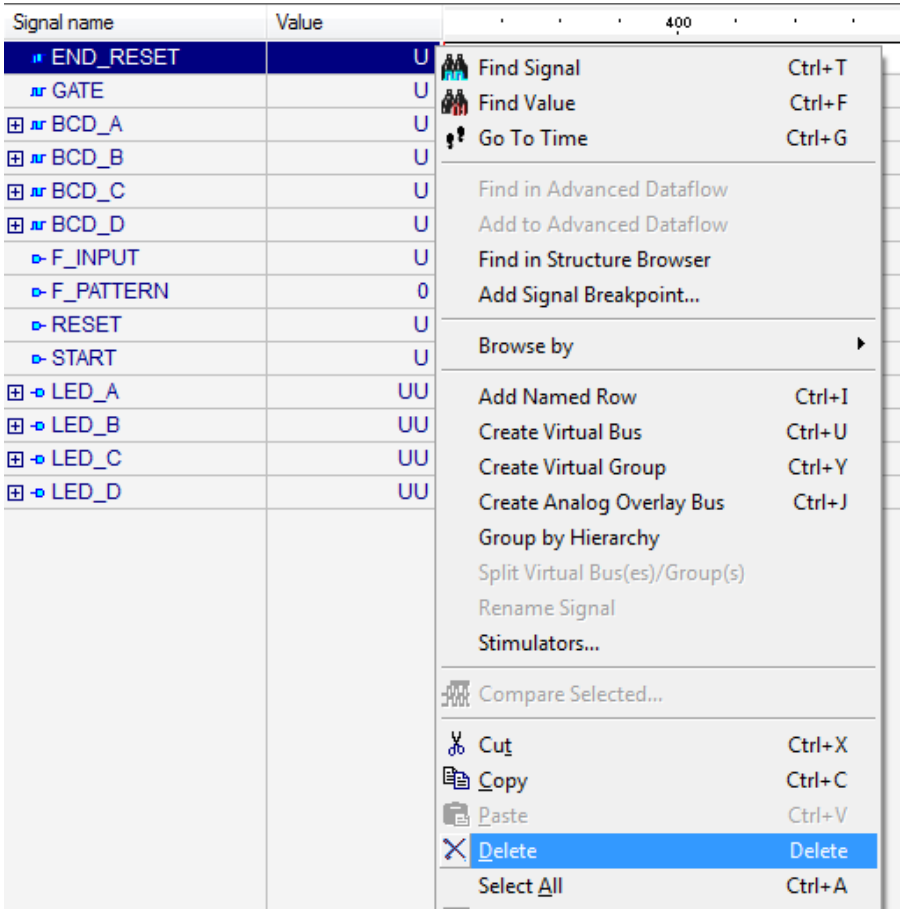
Adding Signals

- To add signals from structure tab right click on any component and select **Add to Waveform** or **Add to Waveform Recursively**.
- When you select **Add to Waveform** signals related to that component will be added to waveform.
- When you select **Add to Waveform Recursively** all signals (internal hierarchical) of the design will be added to waveform



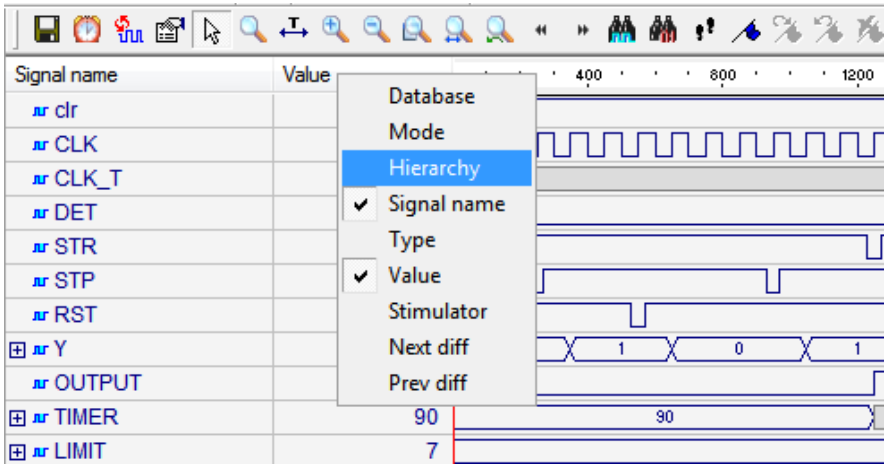
Removing Signals

- To remove the signal from the **Waveform Editor**, select the signal name and press the **Delete** key or choose the **Delete** option from the pop-up menu after right-click of the mouse.
- Select any signal and press the **Del** keyboard key. This signal will be removed from the current waveform window.



Signal Grid

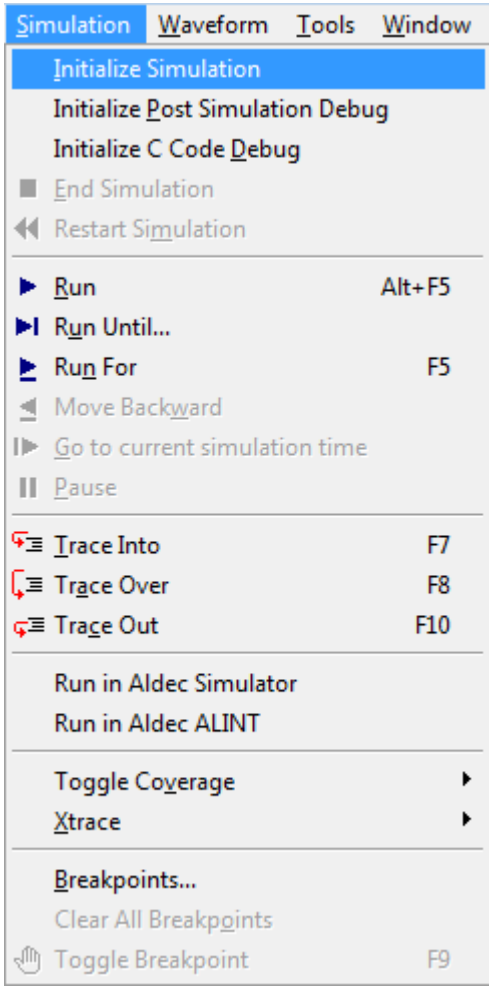
- The Signal Grid panel can be customized to display various information about the signals:
  - Database file name,
  - Mode (signal, input, output etc.)
  - Hierarchy path–Signal name
  - Type
  - Value
  - Next difference
  - Previous difference



- Signals can be sorted according to the content of each column (except Value).Enable the **Hierarchy** column and use it to sort the signals.

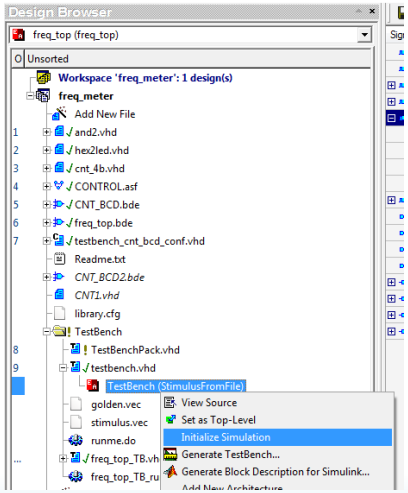
Initializing Simulation

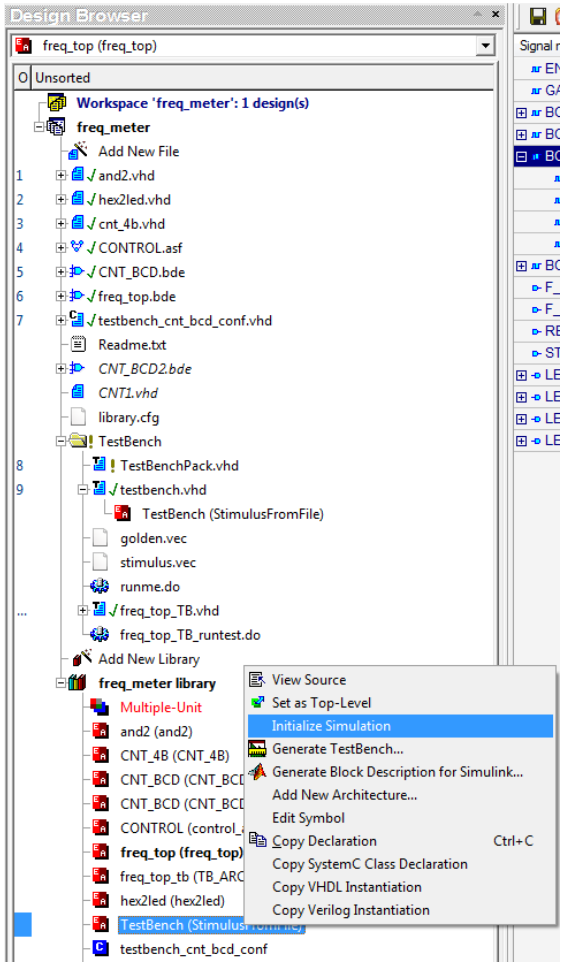
- The **Initialize Simulation** command in the **Simulation** menu launches elaboration and initialization of the simulation model.
- During elaboration, the simulator loads design entities and builds the simulation model in the computer memory.
- During initialization, all objects in the model (signals, variables, etc.) acquire their initial values (either default or explicitly specified) and all concurrent processes are executed once until their suspension.



Initializing Simulation

- You can initialize simulation using right click menu for **Design Browser** window. There are two ways to do this:
  - Right click on compiled object from expanded files and select **Initialize Simulation**
  - Right click on compiled object from expanded work **Library** and select **Initialize Simulation**.





Stimuli

- Active-HDL supports the following methods of stimulating or forcing input signals during the simulation:
  - Manually selected stimulators from the Active-HDL resources
  - HDL Testbench files – see Course 6 Testbench for details
  - Simulation commands entered in the **Console window**
  - Files containing simulation macro commands
  - Test Vector files imported from Active-CAD
  - Simulation input based on waveforms edited by the user

Stimulators

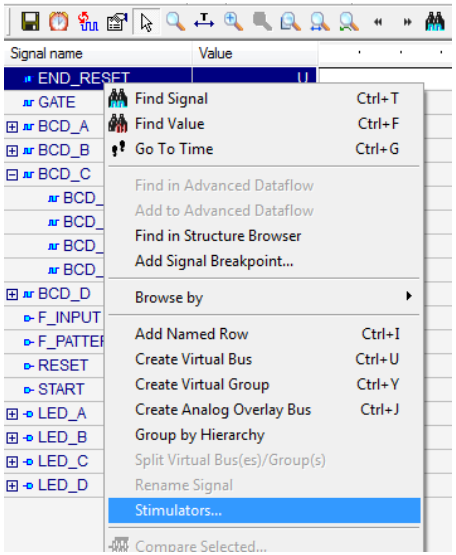
- For the purpose of this course, we will use stimulators to drive the model with its required stimuli.
- **Stimulators** are specialized signal waveform generators that can produce any desired legal value on the model's inputs. There are several types of stimulators:
  - Clock
  - Formula
  - Value
  - Hotkey
  - Counter
  - Predefined
  - Custom
  - Random

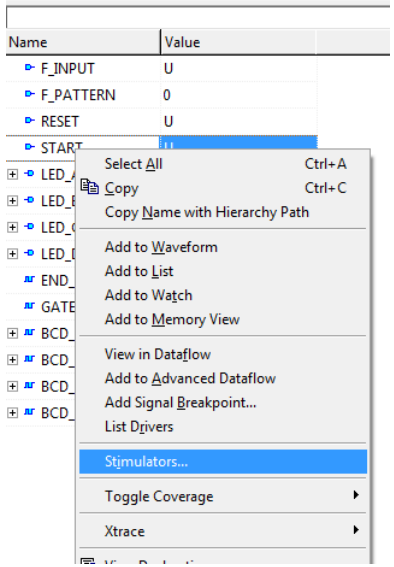
Stimulator types

- **Clock** drives a signal with a clock pulse wave.
- **Counter** drives a signal with a sequence of values that represent consecutive states of a counter.
- **Custom** drives a signal with its own waveform existing in the Waveform Editor window.
- **Formula** drives a signal with values defines by a formula expression.
- **Hotkey** drives a signal with values toggled with a specific hotkey.
- **Predefined** specifies that the signal is to be driven with a named stimulator whose definition is on the Predefined tab.
- **Value** drives a signal with a constant value.
- **Random** drives a signal with a sequence of integer values distributed according to standard probabilistic functions.

Opening GUI Stimulator Window

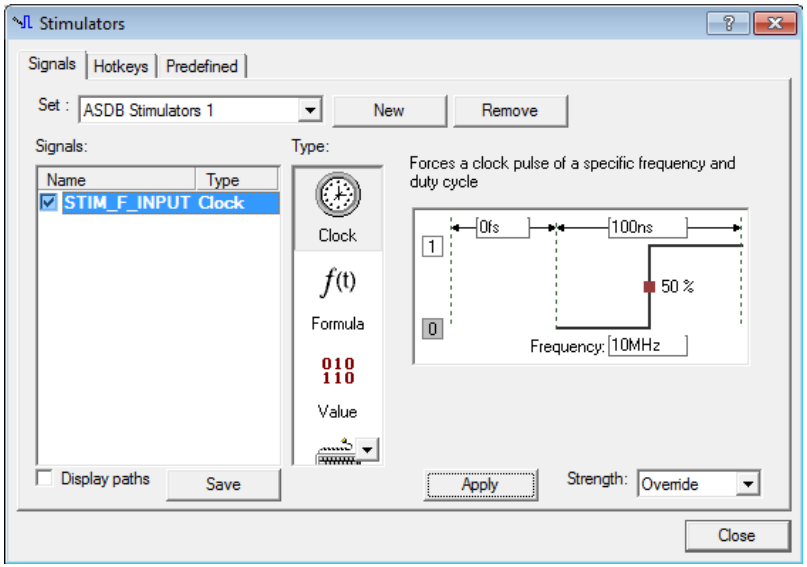
- To open GUI **Stimulator** right click on any signal inside waveform and select **Stimulator…** option from context menu.
- To add signals (inputs) into graphical stimulator click on desired signals inside waveform, all selected signals will be added to GUI Stimulator.
- You can also call Stimulators by choosing the **Stimulators…** option from the pop-up menu of **Object Viewer** window from **Structure** tab of **Design Browser**.

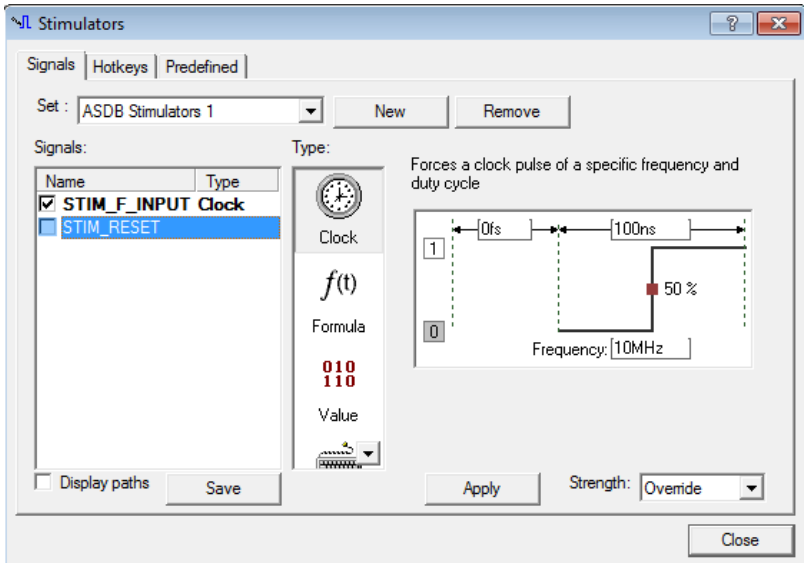




Assigning Stimulators

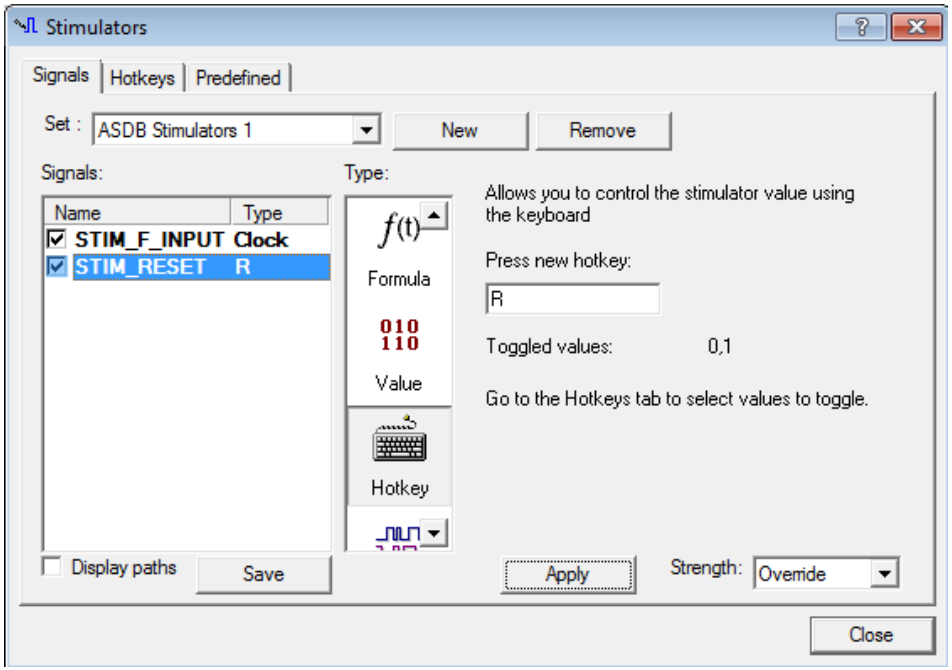
- Choose the Clock type stimulator and click the Apply button to accept the default settings. To assign the next stimulator, you do not need to close the Stimulators window. Move it aside and select the reset signal in the Waveform Editor. Notice the reset signal appear in the Stimulators window.



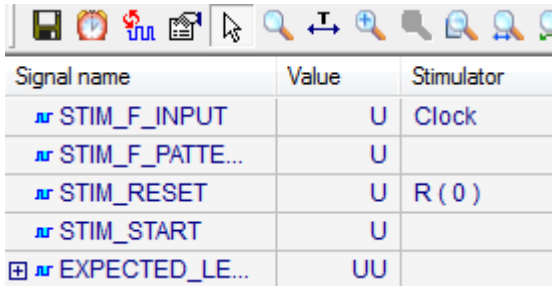


Assigning Stimulators

- Use the icons to scroll the stimulators types in the window. Select the **Hotkey** stimulator type. Click inside the **Press new hotkey** field and press the **R** keyboard key. Accept the stimulator by clicking the **Apply** button.



Notice the tool displays the stimulator type in the **Stimulator column.**



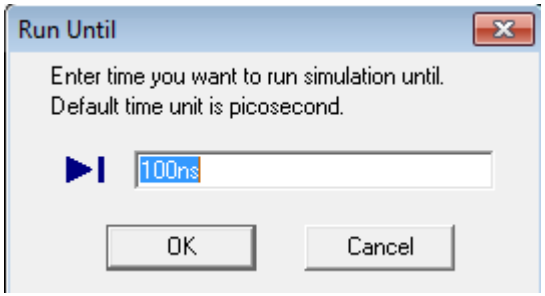
Signal name	Value	Stimulator
STIM_F_INPUT	U	Clock
STIM_F_PATTE...	U	
STIM_RESET	U	R (0)
STIM_START	U	
EXPECTED_LE...	UU	

Advancing Simulation

- When the signals have stimulators assigned, you can run the simulation. Active-HDL lets you advance simulation by clicking the three tool bar buttons. 100 ns
- The **Simulation | Run** command runs simulation for an unspecified amount of time. The simulation stops when either of these conditions is met:
  - There are no more test vectors.
  - A breakpoint in the code has been set.
- In each of the above cases, you can break the simulation by using the **Simulation | Pause** command or by clicking
- The **Simulation | Run For** command advances simulation by a specified time step. To set the time step, use the **Simulation Step** box located on the main toolbar: 100 ns
- The **Simulation | Run Until** command advances simulation until a specified time point.

Advancing Simulation

- Set the simulation step to 50 ns in the **Simulation Step** box. 50 ns
- Click the **Run for** button or press the **F5** keyboard key to advance the simulation.
- Now, press the **R** keyboard key.
- Click the **Run until** button and set the time to 60 ns.



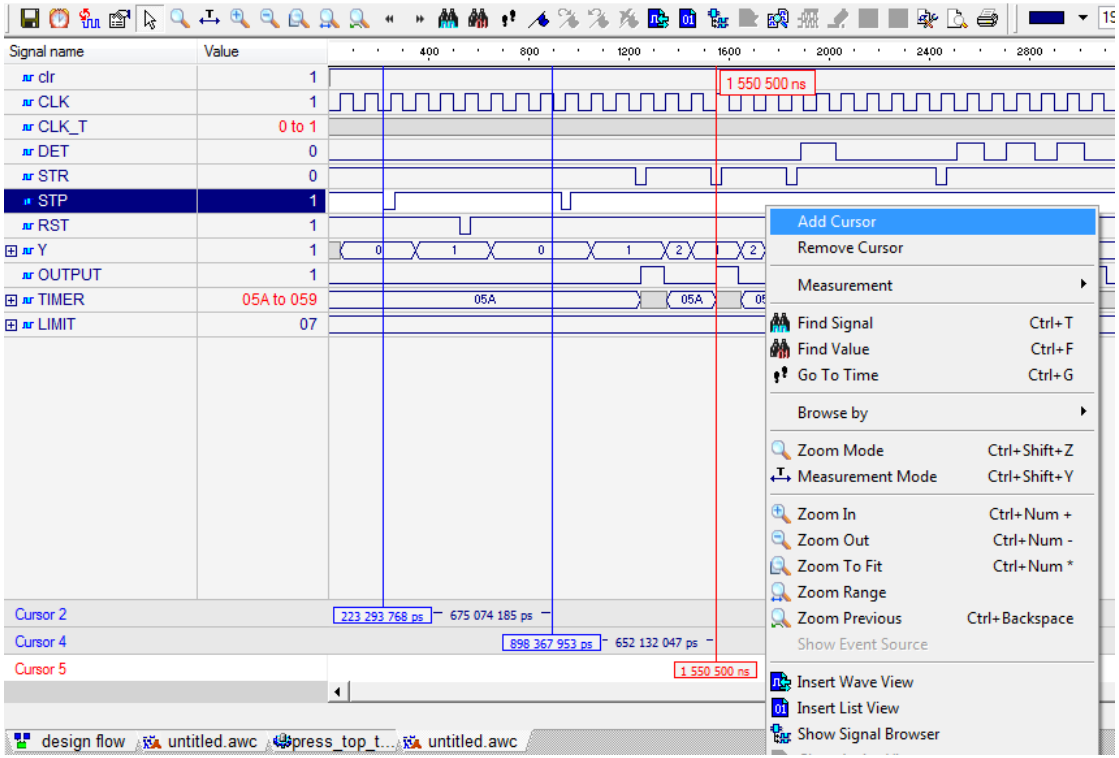
- Press the **R** key again.
- Set the simulation step to 100 ns in the **Simulation Step** box.
- Click the **Run** button and after a couple of seconds.
- Click the **Zoom In** button several times.
- Click the ***'plus'*** sign next to BCD_D bus to expand the signal.

Customizing the View - zoom

- The **Waveform Editor** allows scrolling the display with simulation results. To zoom the display that best suits the view, you can use toolbar buttons or the corresponding options from the **View** menu:-
  - To increase the zooming factor twice or choose **Zoom | In**.
  - To decrease the zooming factor twice or choose **Zoom | Out**.
  - To adjust the zooming factor so as to display the whole timing, or choose **Zoom | Zoom To Fit**.
  - To specify exactly the zooming range. Type the time points in the **Zoom** window.
- You can also switch to Zoom mode by clicking the **Zoom mode** button. In the zoom mode, you can select manually the zoom scope and increase it. To decrease the zoom, press and hold down the **Ctrl** key.

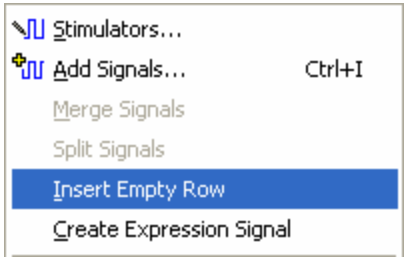
Customizing the View - Cursors

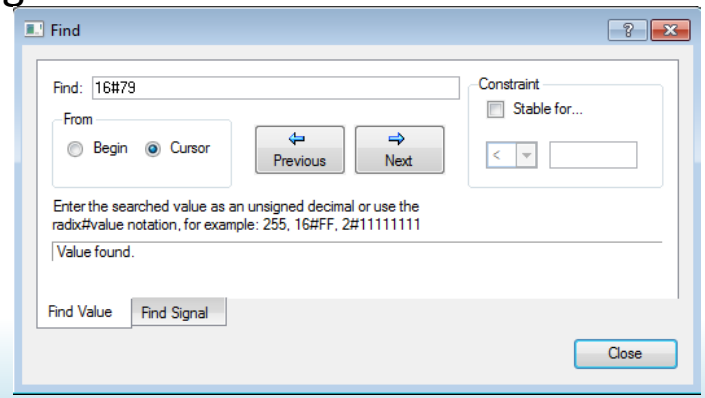
- An arbitrary number of cursors is available to the user. A cursor can be created at any time and given a name that suits you. The Accelerated Waveform displays distance between each two cursors.
- Clicking a given cursor makes the cursor active (displayed in red).



Customizing the View

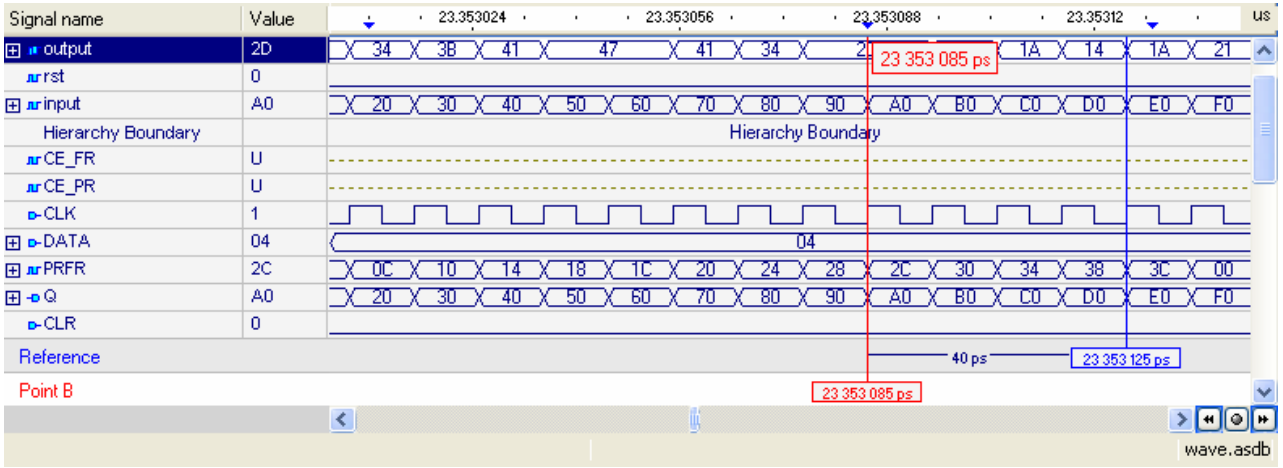
- Choose the **Colors** option from the **Accelerated Waveform** section of **Tool | Preferences** menu for different colors of the waveform.
- Waveform Editor offers many useful functions for browsing and searching the results.
  - Go to a specified time point
  - Set and browse simulation bookmarks
  - Right click on the signal and choose **Insert Empty Row**.
  - Search for a signal value and text in comments.





Customizing the View - Bookmark

- The bookmarks can be used to mark particular places on the waveform ruler and then move the current cursor from one bookmark to another. The number of bookmarks is unlimited. Controls for bookmark navigation are:
  - Toggle bookmark
  - Remove all bookmarks
  - Previous bookmark
  - Next bookmark

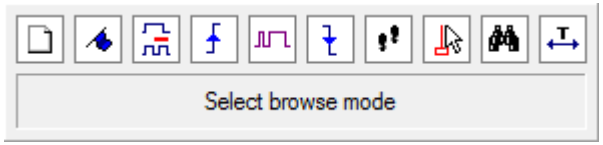


Customizing the View

- The **Waveform Viewer/Editor** allows you to navigate backward and forward through events.
- If there are no signals selected in the waveform window, an event search goes through all signals.
- After selecting one or more signals in the waveform, the timing cursor jumps to the nearest event found among displayed in the waveform window signals.

Customizing the View

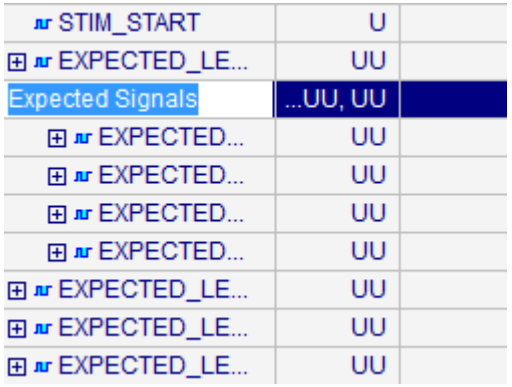
- Click the Select Browse Object button located in the bottom-right corner of the Waveform Editor window.
- Select any browsing mode form the browse selection box.

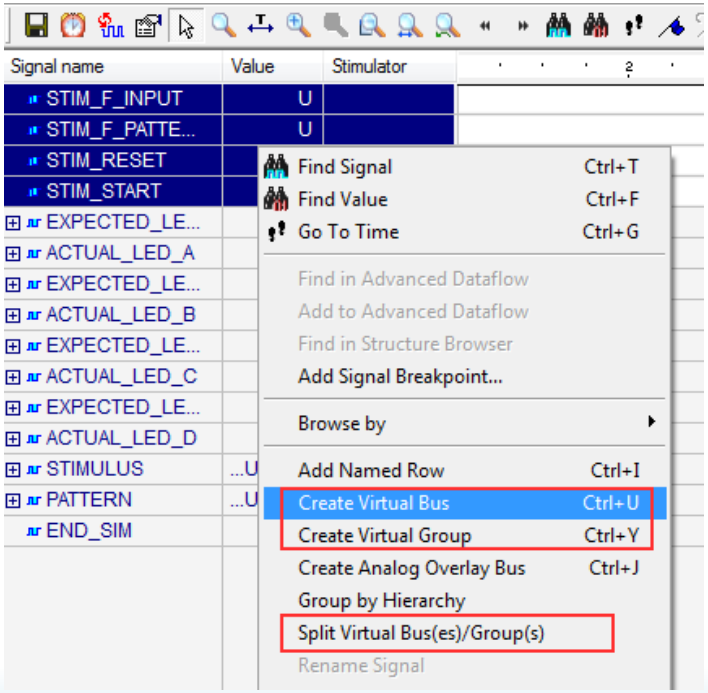


- The waveform editor allows grouping and collapsing signals to/from a virtual bus. This can be done by right-clicking the mouse in the Name column of the Waveform Editor and choosing Create Virtual Bus / Create Virtual Group Signals/ Split Signals.
- Aggregated objects (virtual buses) can be also renamed in any of the available working modes (Select, Zoom, Measurement, or Edit).

Customizing the View

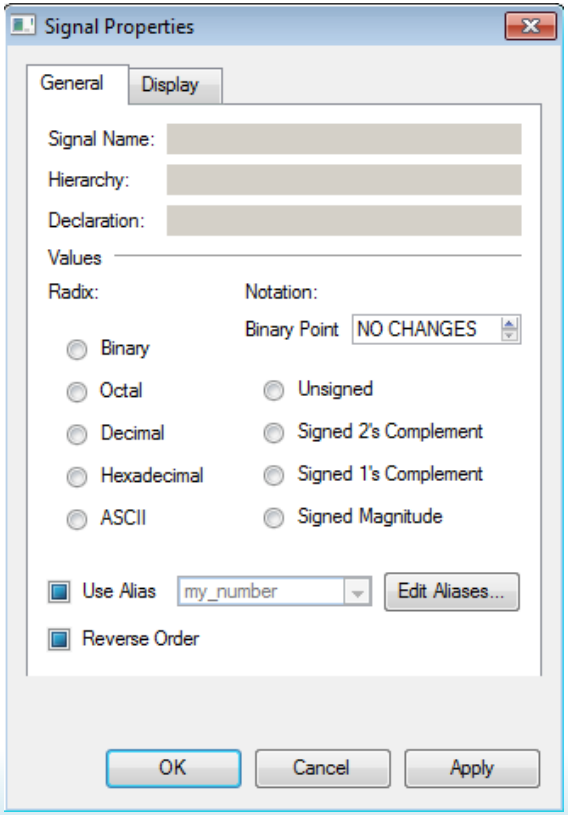
- You can create another virtual group inside main virtual group.
- Drag-n-Drop signals inside group to add signals.
- Use **Split Virtual Bus(es) / Group(s)** option to split groups and buses.





Signal Properties

This dialog box is used to view properties of the selected signal in the Waveform Viewer/Editor window. To open the dialog, select the desired signal with the right mouse button and then choose Properties from the shortcut menu.

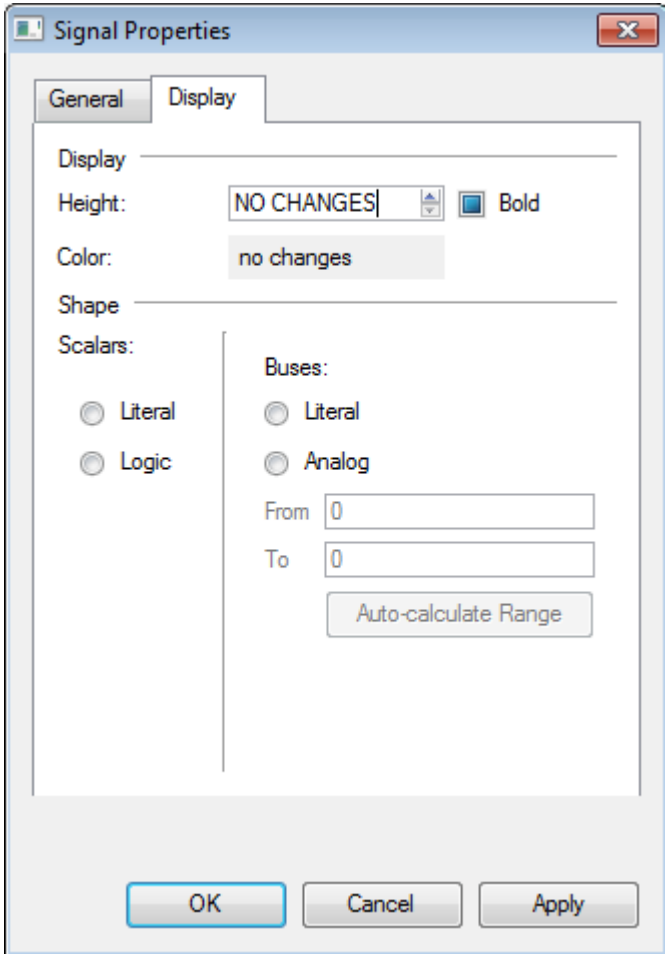


- This dialog box displays the signal name, its hierarchy, and its declaration.
- You can choose the signal to display the values with a Radix of 2(binary), 8(octal), 10(decimal), 16(hexadecimal).
- Use Alias enables the use of alias selected from the list box.
- Different notations can be used to display the values of signals like Unsigned, Signed 2's complement, Signed 1's complement, Signed Magnitude.

Signal Properties

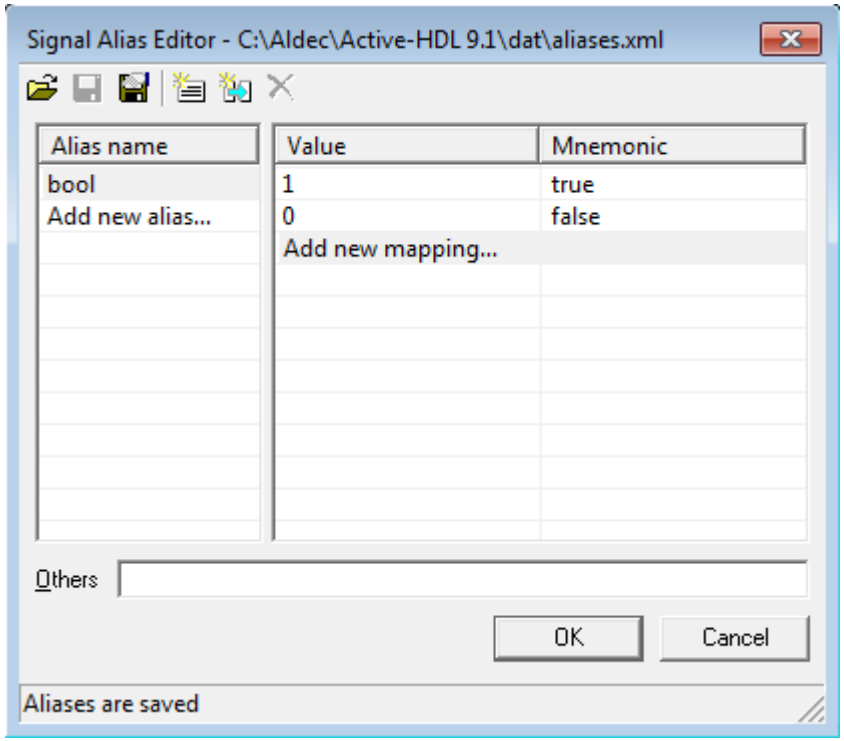
Auto-range calculation for the display of analog signals:

- This feature allows to calculate automatically the value range of the analog signal displayed in the waveform window.
- The Auto-range feature searches for the minimum and maximum value of the displayed waveform and automatically adjusts its range to present the entire signal.
- If you choose Analog and the signal whose properties you are about to change has already been displayed, the auto-range feature will calculate the proper range values and display them in the from and to edit boxes.



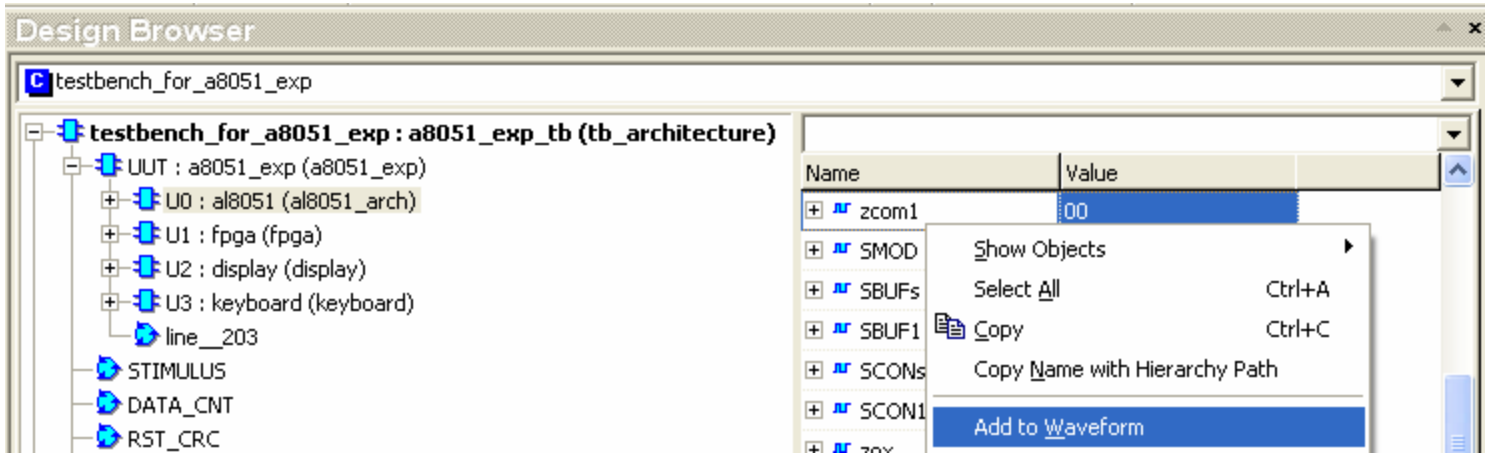
Signal Alias Editor

- The Signal Alias Editor is designed for creating and modifying signal aliases.
- It can be used to:
  - Create a new alias
  - Create new value mappings for existing aliases
  - Modify or delete existing aliases and their value mappings



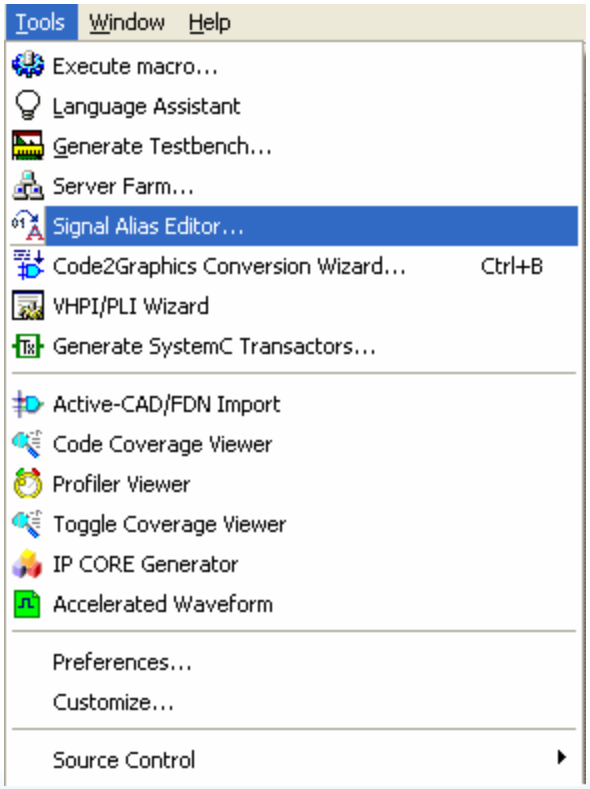
Signal Alias Editor

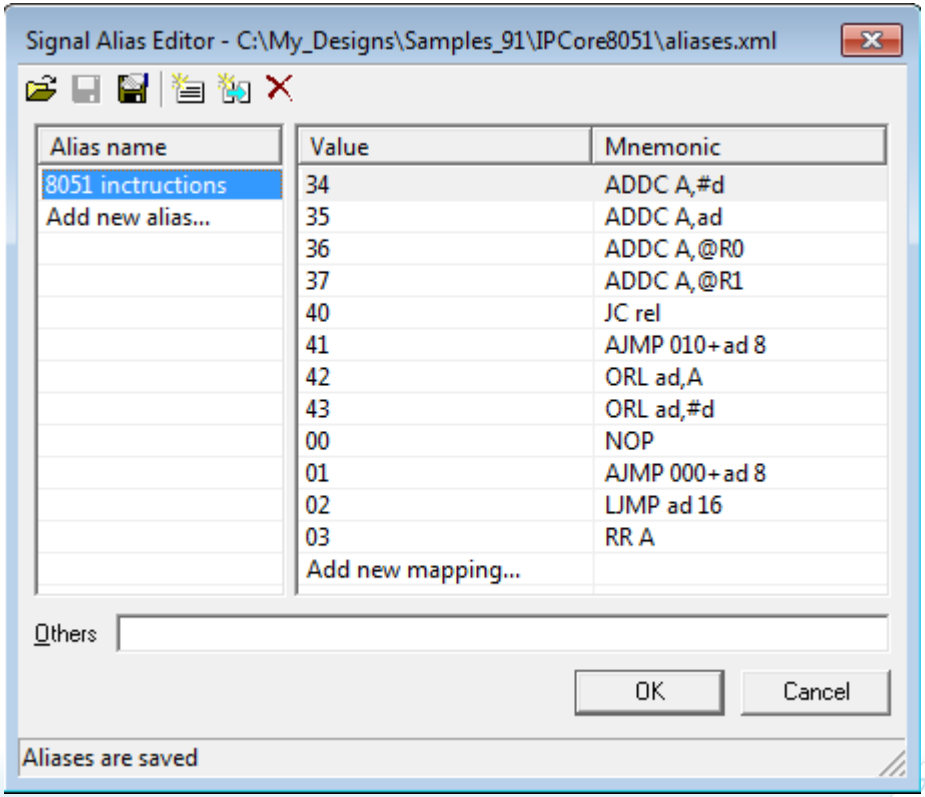
- Open the *IPCore8051* design.
- Click **Compile All with File Reorder** button
- Select *testbench_for_a8051_exp* configuration as a top-level unit and press OK button.
- Initialize simulation and add /UUT/U0/zcom1 signal to the waveform:



Signal Alias Editor

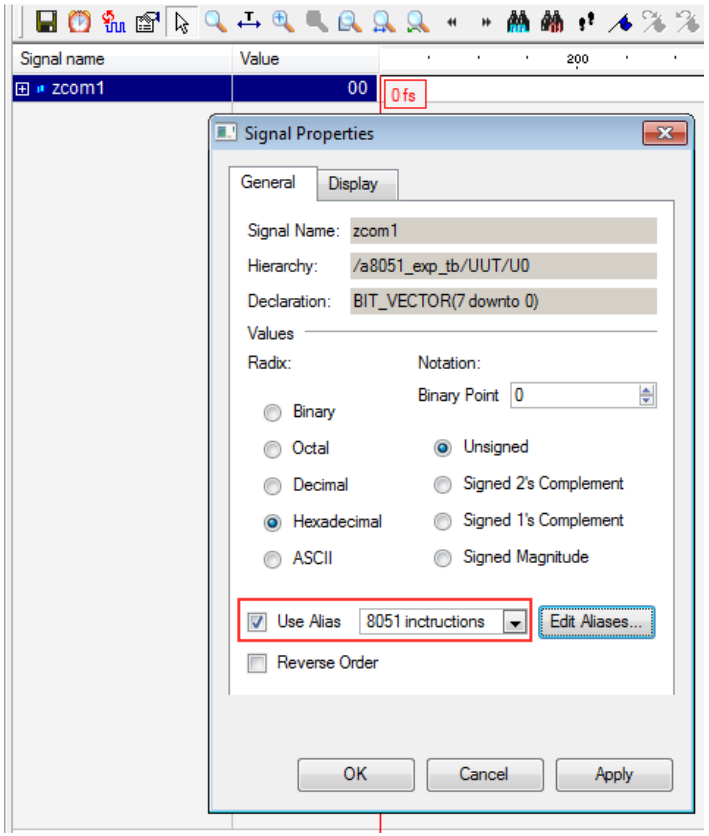
- Choose the **Signal Alias Editor** option from the **Tools** menu. Create aliases for 8051 instructions and save them to a file using **Save As** toolbar button

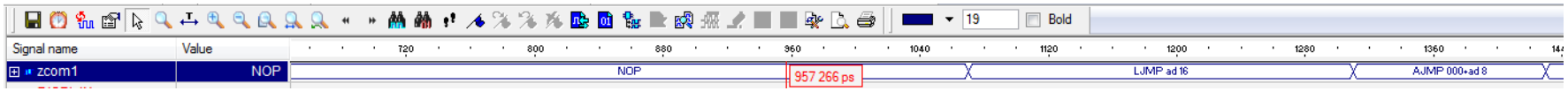




Signal Alias Editor

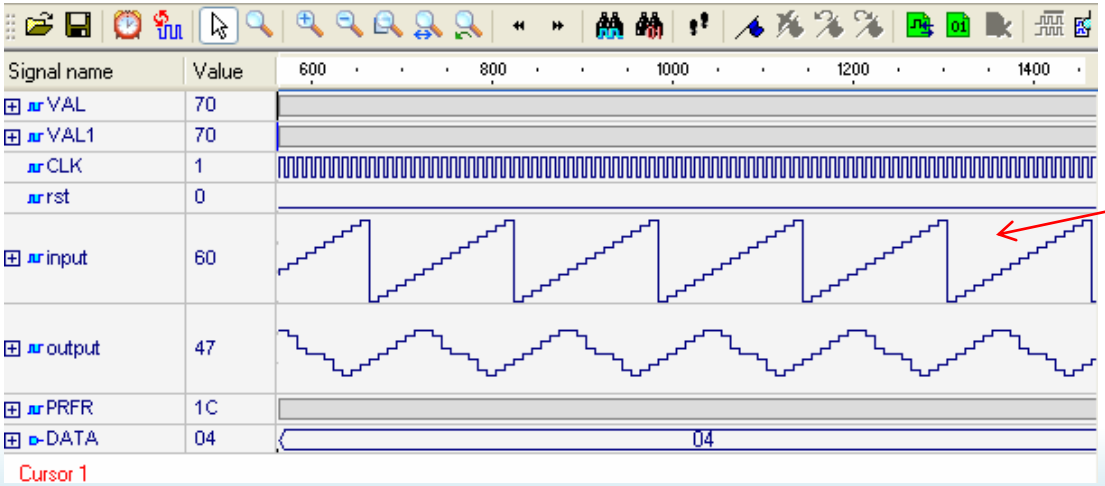
- Choose the **Use Alias** option in the **Properties** window for zcom1signal.
- Run simulation and in the waveform you will see aliases mapped to the signal values.

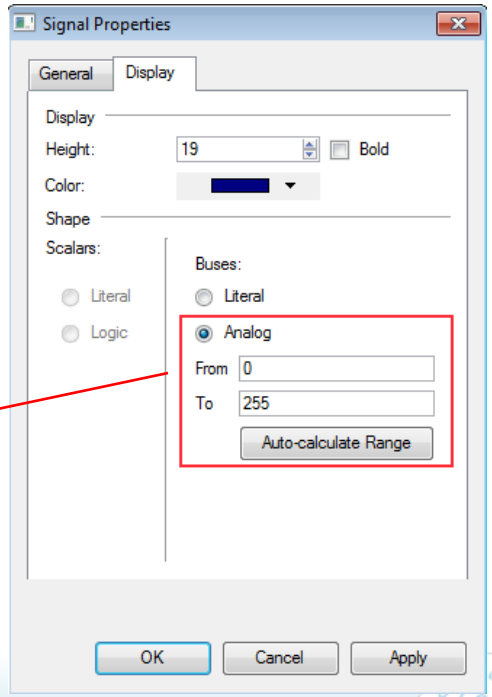




Analog Display

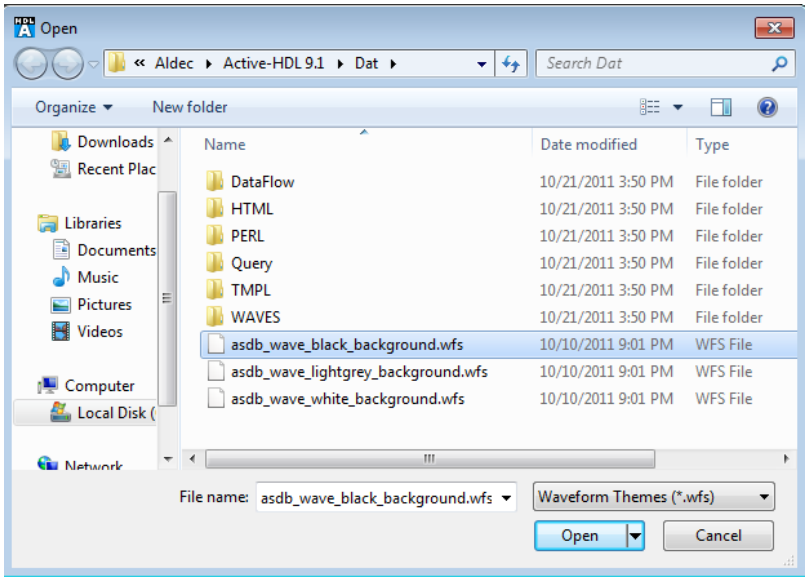
- The Accelerated Waveform supports analog display of signals. In this mode, signal values are treated like samples forming an analog waveform.
  - To enable the analog display, call the Signal Properties dialog using context menu option or use Alt+Enter key combination.
  - Switch to the Display tab.
  - Choose Analog in the Shape section.
  - Adjust the Height.

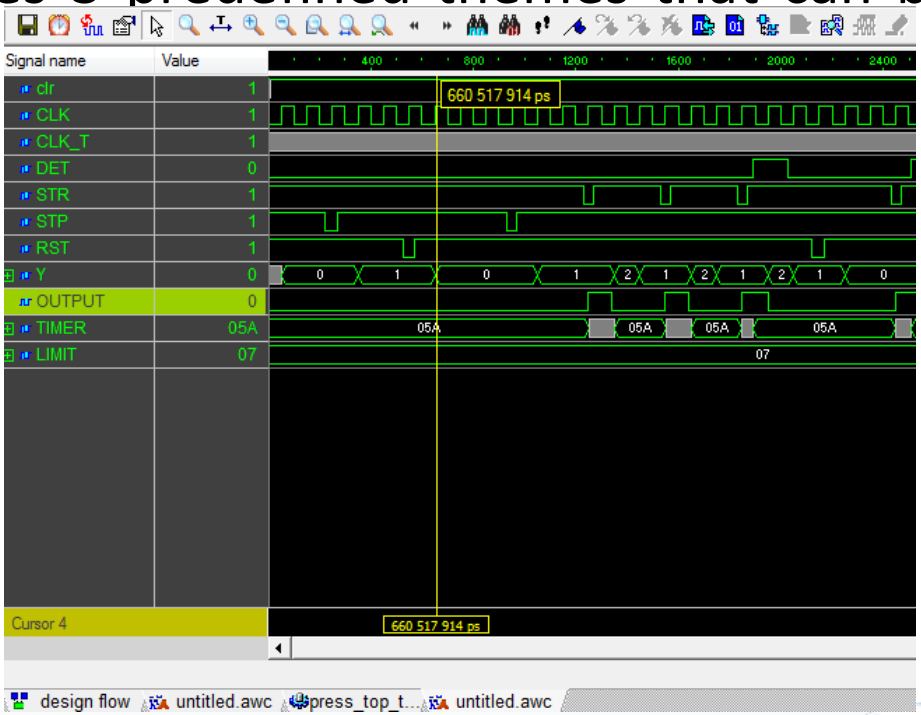




Appearance Customization

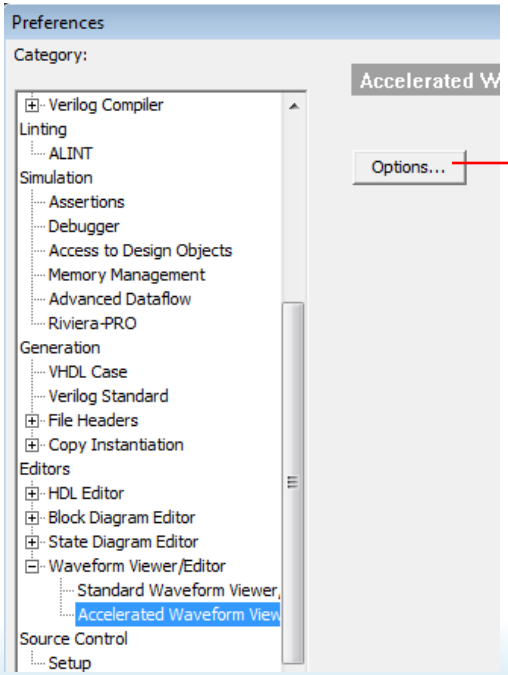
- The Accelerated Waveform Viewer appearance settings can be customized to suit the user's visualization preferences. Changes made to the colors, fonts and sizes can be then saved to a file. Active-HDL installation provides 3 predefined themes that can be selected.

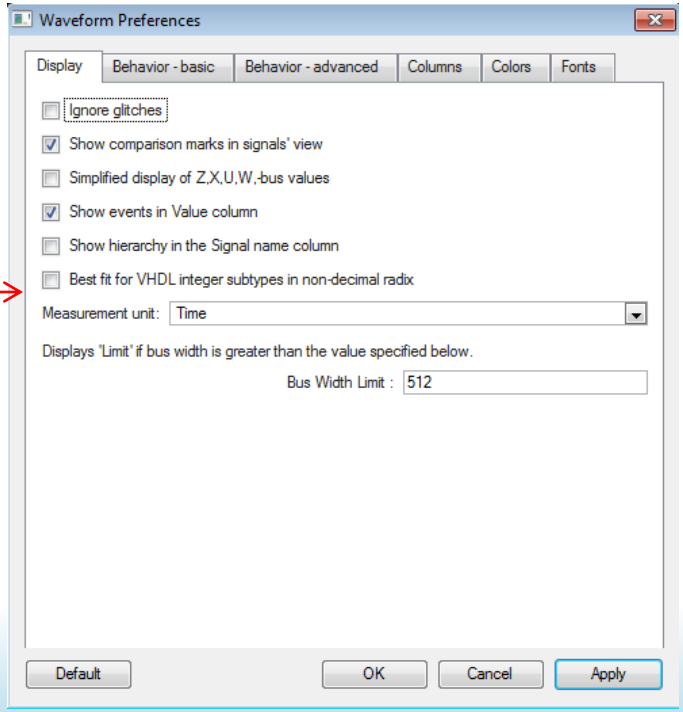




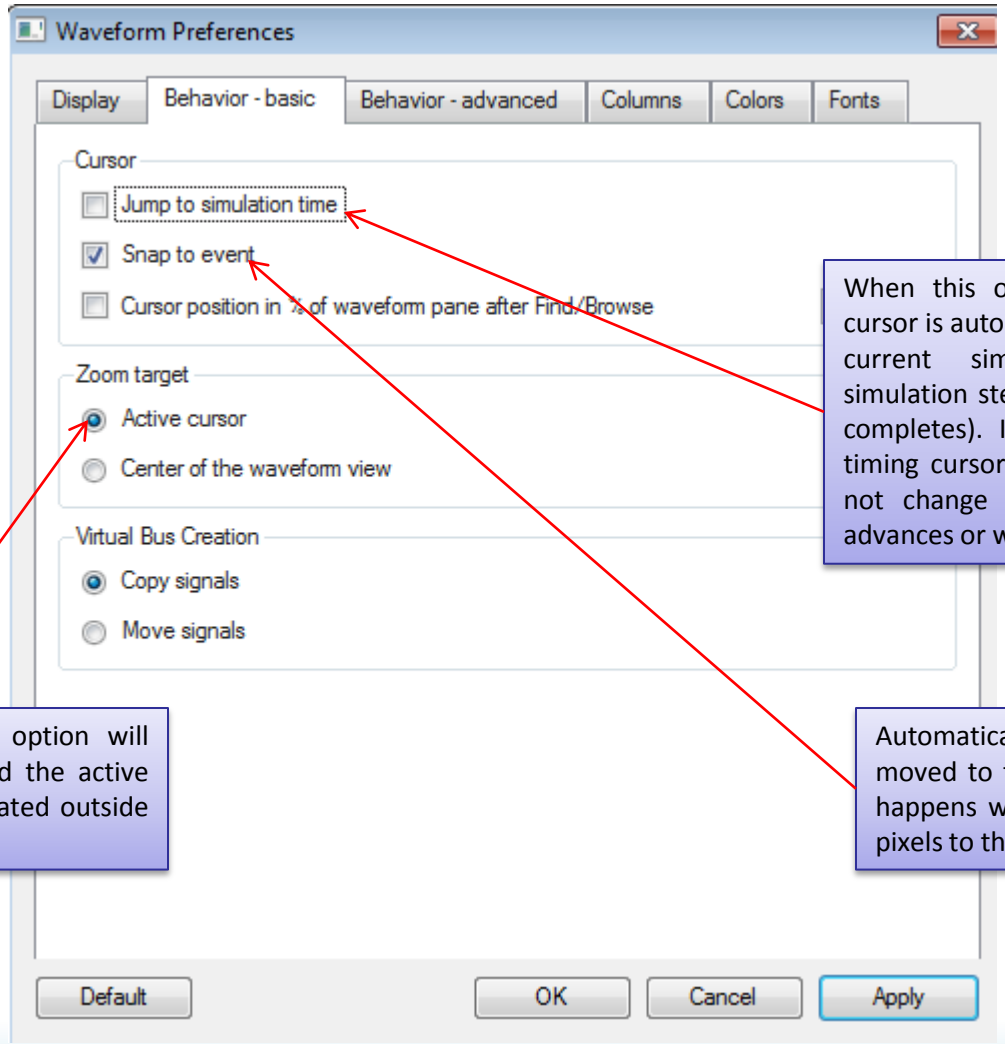
Waveform Preferences

- This dialog box allows you to set display properties of the Accelerated Waveform Viewer. To open the dialog box, click the **Waveform Preferences** button in the Accelerated Waveform Viewer window or click on the Options button in the Accelerated Waveform Viewer tab in the **Tools |Preferences** dialog box.





Waveform Preferences | Behavior-basic



When this option is checked, the timing cursor is automatically synchronized with the current simulation time after each simulation step (or when the run command completes). If this option is disabled, the timing cursor stays in its position and does not change the location after simulation advances or waveforms are redrawn.

When selected, the **Zoom In** option will zoom in the time range around the active cursor, even if the cursor is located outside the viewing area.

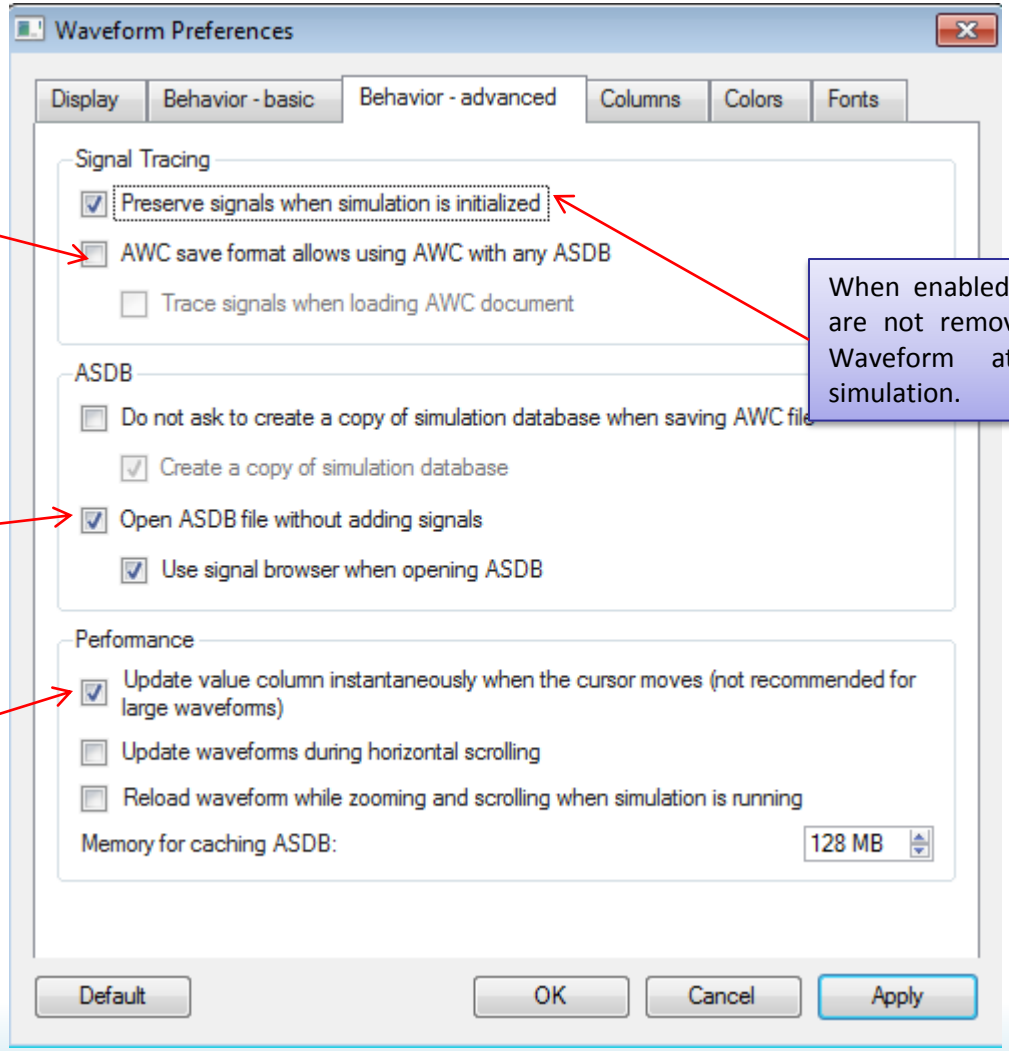
Automatically aligns the cursor that is being moved to the nearest event. The alignment happens when an event is located within 5 pixels to the cursor.

Waveform Preferences | Behavior-Advanced

Allows you to re-use the Waveform View Configuration files (*.awc) in subsequent simulation runs after you close the last simulation session.

Opens an empty waveform file even if there are recorded signals in the simulation database (*.asdb).

Updates values in the **Value** column as the cursor is being moved.



When enabled, previously added signals are not removed from the Accelerated Waveform at the initialization of simulation.

Saving the waveform

- Waveform Editor saves simulation results into waveform files (AWF) that are based on a text format. The editor also allows you to export waveforms into other format files. The following text formats are supported:
  - WAVES-compliant test vector files (*.vec). This format is briefly referred to as VEC.
  - Regular VHDL code with process statements generating equivalent signal waveforms (*.vhs).
  - Verilog Initial Block with statements generating equivalent signal waveforms (*.ver).
  - Verilog Value Change Dump, Extended VCD, Xilinx Xpower VCD Compatible(*.vcd). This format is described in Verilog standard.
  - List Viewer (*.lst) and Tabular Format (*.exp).
  - Chip Express Test Vector (*.ctv).
  - Additional supported formats: TSSI and Macro

Note: Files of these formats may be used for testbench generation. For details see Course 6 Testbenches.

Exporting the waveform

- Choose Export | Waveforms from the File menu.

  the Save As dialog box will open.
- In the Save as type box, select the desired format type. The following formats are supported: VHS, VEC, VER, VCD, EXP, LST, CTV,TSSI and Macro.
  - In the Range box, select which signals are to beexported. You can choose one of the following:
  - All signals
  - Selected signals (you should select the desired signals before using the Export Waveforms command).
  - All ports
  - Input ports
- Specify the file name, the folder to which the file will be saved, and then click Save.

Exporting the waveform

File Edit Search View Workspace Design Simulation Waveform Tools

Open Workspace/Design Explorer
Close Workspace
New
Open... Ctrl+O
Open design from Source Control
Open Symbol...
Close Ctrl+F4
Close All
Save Ctrl+S
Save As... F12
Save All
Save Design As...
Send
Import
Export
Page Setup...
Print preview...
Print File... Ctrl+P
Print Design Files... Ctrl+Shift+P
Print Workspace... Ctrl+Shift+W
Recent Files
Recent Workspaces
Exit Alt+F4

Waveforms
HTML...
PDF...
Graphics...
Design to HTML
Workspace to HTML
Design to PDF
Workspace to PDF
EDIF 2 0 0

Signal name
STIM_F_INPUT
STIM_F_PATTE...
STIM_RESET
STIM_START
EXPECTED_LE...
Expected Signals
EXPECTED...
EXPECTED...
EXPECTED...
EXPECTED...
EXPECTED_LE...
EXPECTED_LE...

Export Wizard

Export Wizard

Export Format
- Export to CTF
- Export to CTV
- Export to LST
- Export to TSSI
- Export to Macro
- Export to VCD
- Export to VEC
- Export to VER
- Export to VHS

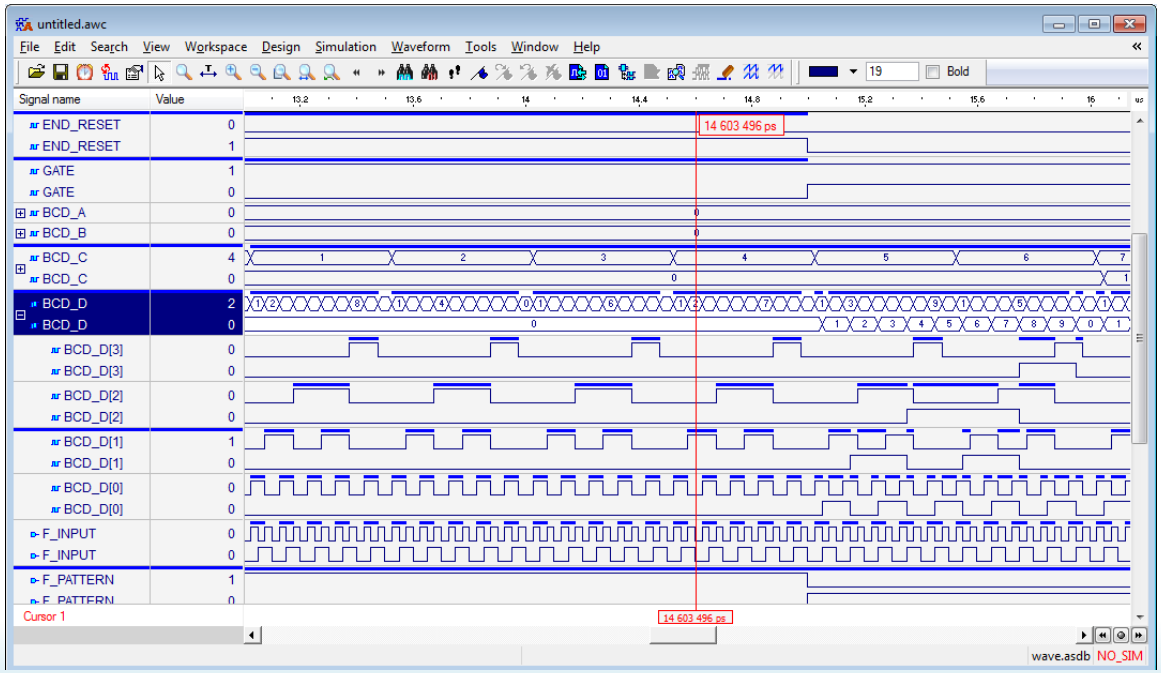
Signal Filter
Selected signals

File Name
c:\My_Designs\Samples_91\Freq_meter\src\untitled
Browse...

< Back | Next > | Cancel

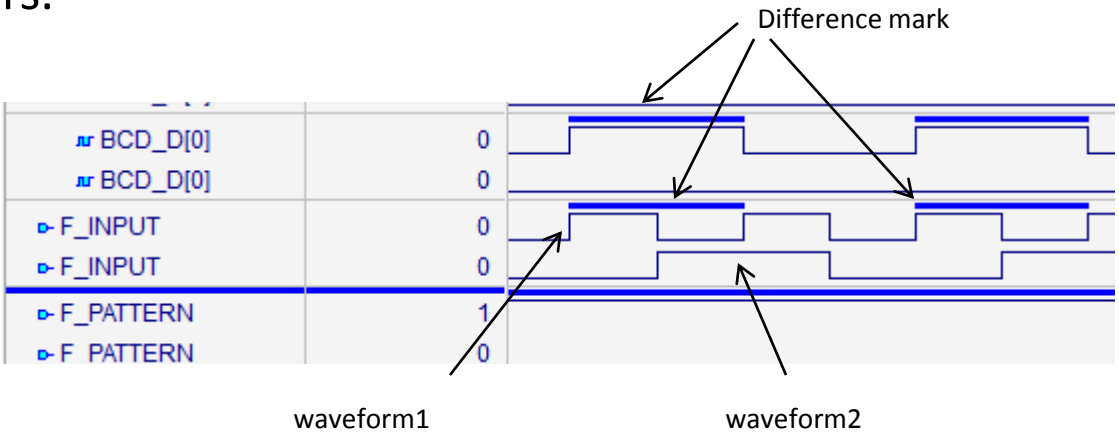
Comparing Waveform

- The Compare Waveforms command compares waveforms displayed in the Waveform Editor window (referred to as original waveforms) with pattern waveforms from a specific waveform file. Once you have made a comparison, you can apply changes to the original waveforms or ignore them.



Comparing Waveform

- To compare waveforms do the following:
  - Choose Compare Waveforms from the Waveform menu. The Open dialog box will appear.
  - Find and select the waveform file where you saved the results and click Open. The pattern waveforms will be loaded into the Waveform Editor window and displayed in red (the original waveforms are displayed in blue). Differences between the original and pattern waveforms are denoted with blue bars.



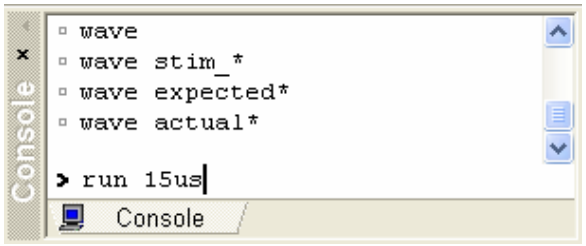
Comparing Waveform

- You can compare signals from the currently loaded waveform with signals in documents of any type that can be imported into the Waveform Editor/Viewer, i.e. *.awf(waveform files), *.vcd(Value Change Dump format files), and *.lst(waveform list files).
- You can enter the range of the comparison time by using the **Range from** and **to** fields. Then you compare signals in the requested range.
- You can define **Offset** for a pattern during the signal comparison. This feature may be very helpful in the analysis of simulation results.
- **Tolerance** allows for specifying an acceptable difference between a signal and corresponding pattern.
- **Source Hierarchy** and **Target Hierarchy** allow comparison of signals from different hierarchy levels.
- **Maximum number of differences** can be precisely specified. If it is exceeded, the comparison will be stopped immediately.

Using Macro command for Simulation

- Active-HDL provides a macro command language for manual entering of such simulation commands as forcing signal values, assigning formulas and executing simulation steps.
- You can force a value on a signal at any time during simulation by entering the appropriate macro command in the Console window. You can also use macro commands to add forced signals to the Waveform Editor, etc.

```
9   acom "$DSN\src\TestBench\TestBenchPack.vhd"
10  acom "$DSN\src\TestBench\testbench.vhd"
11  acom "$DSN\src\testbench_cnt_bcd_conf.vhd"
12
13  asim testbench_cnt_bcd_conf
14  wave
15  wave stim_*
16  wave expected*
17  wave actual*
18  run 25 us
19  endsim
```



Timing Simulation

- Timing Simulation steps:
  - Import structural files
  - Import timing constraints –SDF
  - Set the top-level architecture
  - Open Waveform Editor
  - Add the signals
  - Initialize and run simulation
  - Apply stimulators
  - Measure time differences
  - Save simulation run

VITAL and SDF

- Active-HDL simulator can perform timing simulation based on HDL structural netlists, EDIF netlists, and SDF files. These files are created during the synthesis and implementation processes. The simulator provides built-in acceleration for VITAL packages.
- HDL and EDIF netlists contain structural connections between components. SDF (Standard Delay Format) files contain specific timing constraints of a programmable device.
- To simulate such netlists in Active-HDL, you need to add these files to the current design. You can do it by using the **Add New File** wizard. For details, refer to the previous courses.

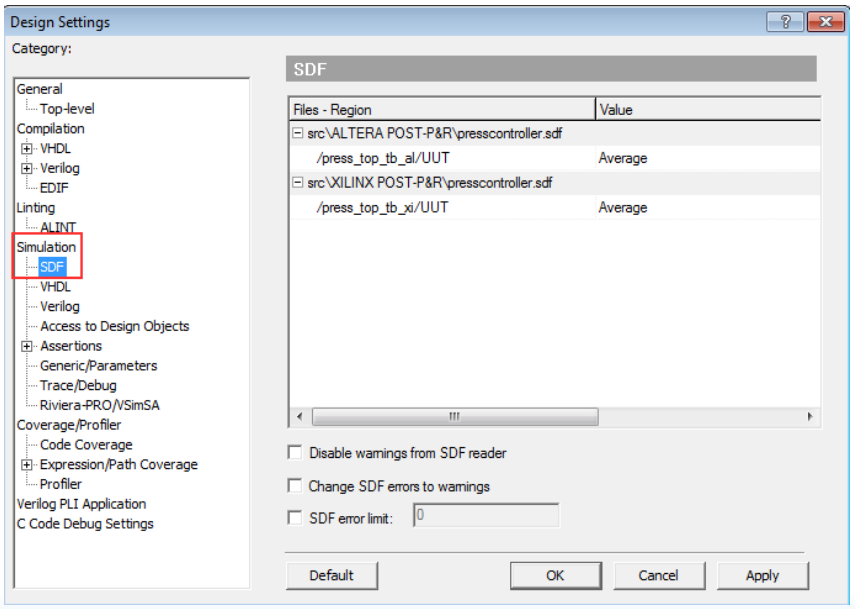
Generics

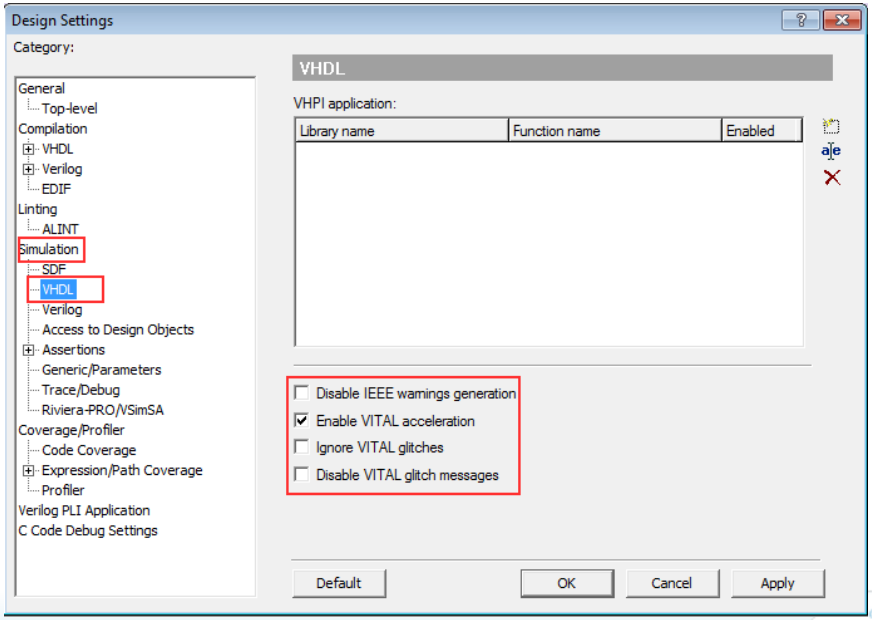
- You can perform timing simulation at any stage of design development. This, however, requires some effort on your side.
- VHDL language supports generics that are timing parameters describing delays in a design. If you created assignments in the model description using generics, you can assign certain values to them externally and, therefore, simulate real programmable device delays.

```
entity Gen_Gates is
generic (Delay : Time := 10 ns);
port (In1, In2 : in Std_Logic;
      Output : out Std_Logic);
end Gen_Gates;
architecture Gates of Gen_Gates is
begin
  . . .
  Output <= In1 or In2 after Delay;
  . . .
end Gates;
```

Timing Settings

- You can set the timing simulation settings in the **Design Settings** window. Here you can specify if the simulator should ignore VITAL glitches for VHDL and specify the type of delays for Verilog. You can also load the SDF file(s) for a specific region to enable timing simulation (not required for Verilog).





Timing Settings

- You can set the timing simulation settings from the macro command file. For this purpose use the *asim* macro command. An example usage is shown below:

```
asim -sdftype /pc_tb/uut="C:/My_Designs/src/Post-P&R/pc.sdf" testbench_for_pc
```

testbench_for_pc

- Specifies the name of the top-level configuration to be simulated.

-sdftyp

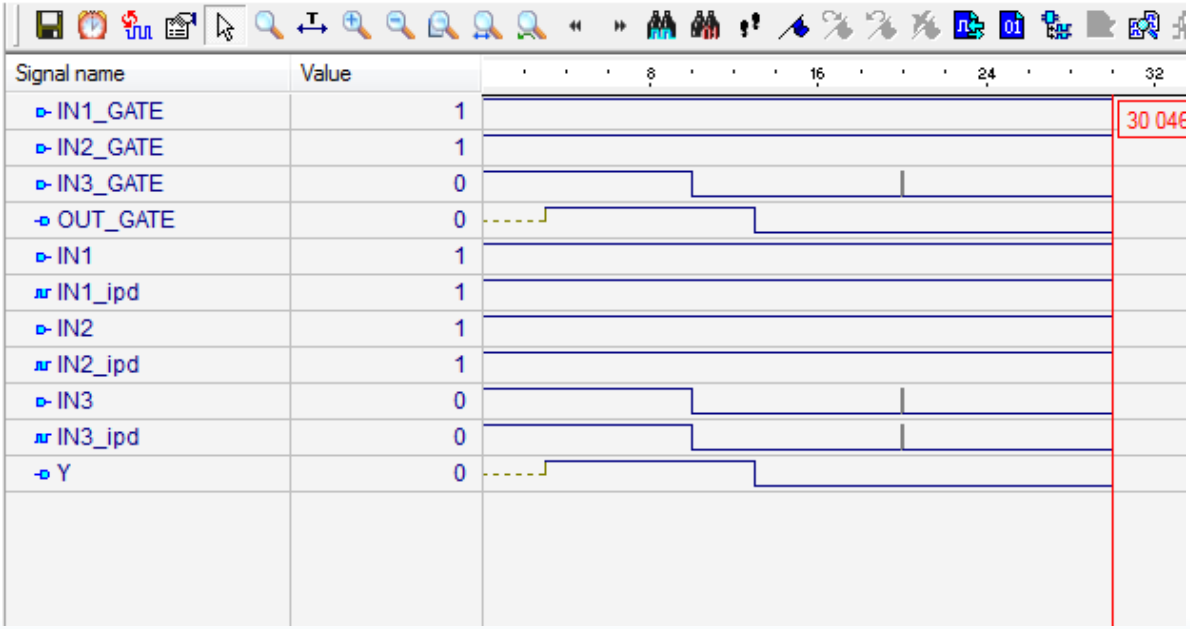
- Annotates VITAL cells in the specified region with typical timing values from the SDF file.

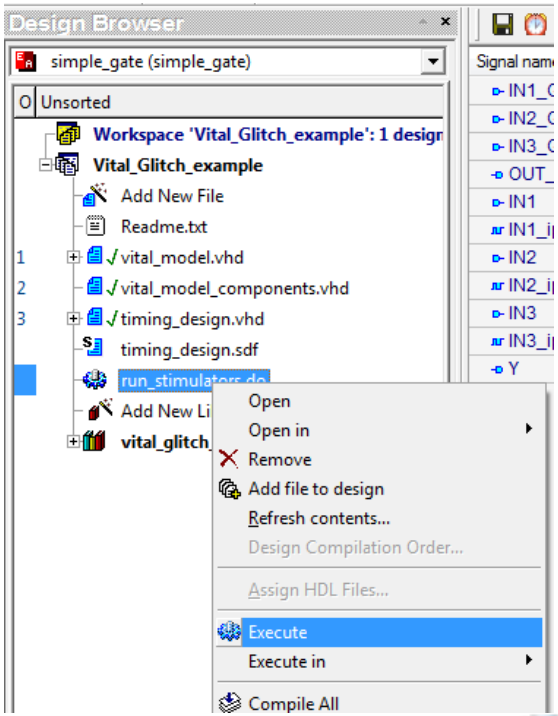
/pc_tb/uut="C:/My_Designs/src/Post-P&R/pc.sdf"

- Specifies the design region into which timing data from the specified SDF file are to be loaded.

Running Timing Simulation

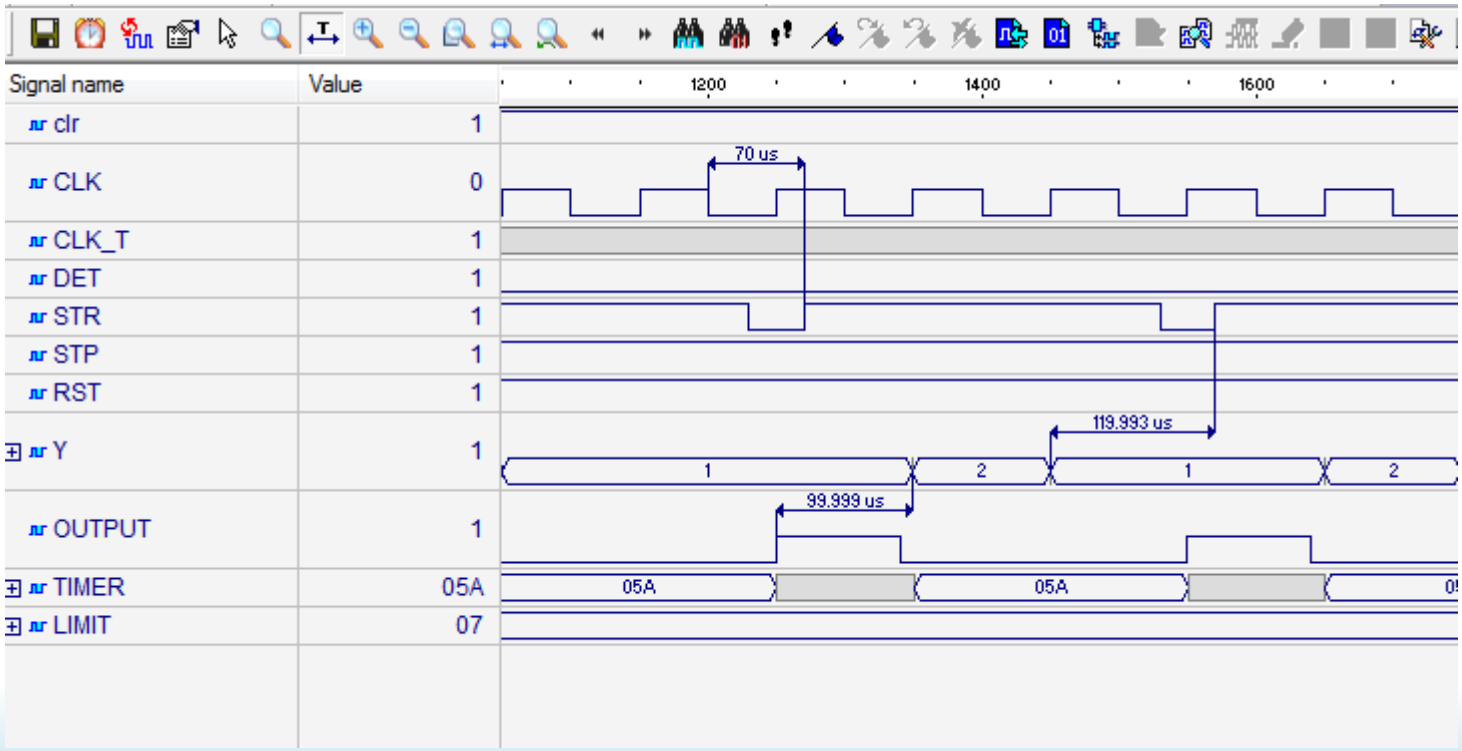
- Open the '*VITAL_Glitch_example*' design.
- Execute the '*run_stimulator.do*' file.
- Observe the simulation results in the waveform.





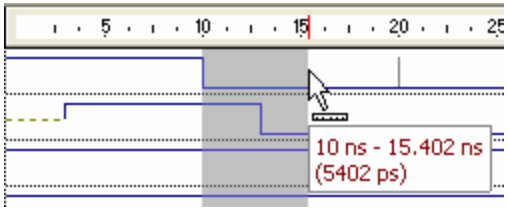
Measuring Distance between Events

- During the timing simulation, the most important issue is to check the timing constraints. Active-HDL facilitates this process providing you with **Measurement Mode** in the **Waveform Editor** window.



Measuring Distance between Events

- Switch to the **Measurement Mode** clicking button.
- Click the event at which you want to anchor one end of the measured area and hold the mouse button.
- Drag the mouse pointer to stretch the measured area (displayed on grayed background) to another event. The time distance is displayed in the tooltip.
- Release the mouse button.



Note: You can set the **Snap to event** option in the **Preferences |Waveform** window to automatically snap the cursor while measuring events.

Tracing Timing Violation

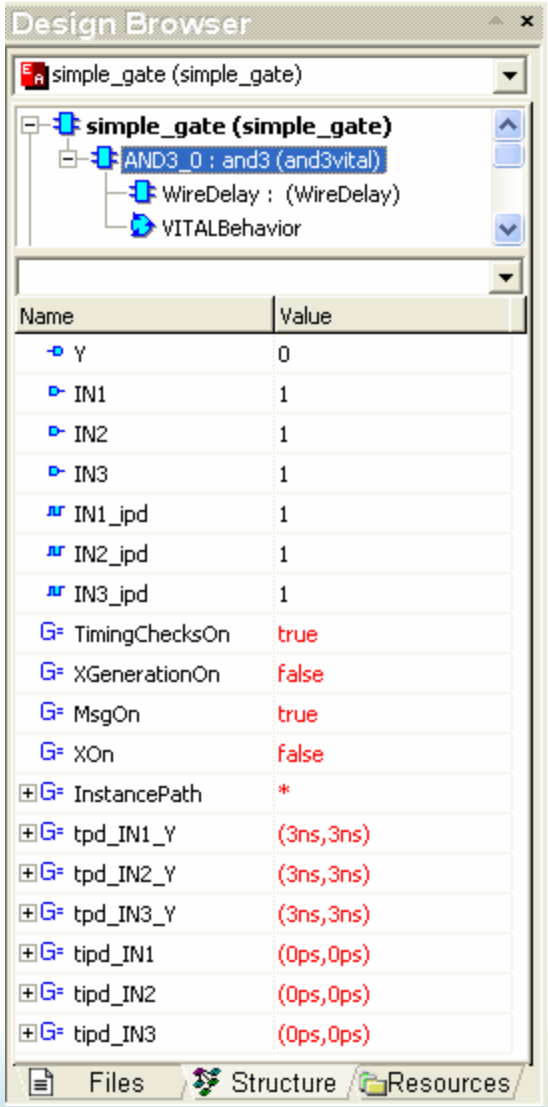
- During the timing simulation (in VHDL), you may observe the glitch warning messages displayed in the **Console** window. An example of a warning is presented below:

```
KERNEL: WARNING: VitalGlitch: GLITCH Detected on port Y ;
Preempted Future Value := 1 @ 23 ns; Newly Scheduled Value
:= 0 @ 23.04 ns;
KERNEL: Time: 20040 ps, Iteration: 1, Instance: /AND3_0,
Process: VITALBehavior.
```

Note: **Glitch** is a short pulse on a signal waveform that is usually undesired and may cause an unexpected design behavior.

Tracing Timing Violation

- To quickly locate the listed instance, switch to the **Structure** tab in the **Design Browser** window. Then, expand the hierarchy tree. Select the *AND3_0* unit, the **Design Browser** displays additional information in the lower part of the window.



Tracing Timing Violation

- As you can see, there are time structures of the *VitalDelayType01* type. Each of these structures contains two time values:
  - first, containing the time for signal transition from 0 to 1
  - second, containing the time for signal transition from 1 to 0
- The tipd_IN1, tipd_IN2, tipd_IN3 and structures hold time values for the input delays. This is the time after which a signal change is propagated from the input to the circuit.
- The tpd_IN1_Y, tpd_IN2_Y, tpd_IN3_Y and structures hold time values for the output delays. This is the time after which signal change is propagated through the circuit to the output.

Name	Value
Y	0
IN1	1
IN2	1
IN3	1
IN1_ipd	1
IN2_ipd	1
IN3_ipd	1
TimingChecksOn	true
XGenerationOn	false
MsgOn	true
XOn	false
InstancePath	*
tpd_IN1_Y	(3ns,3ns)
tpd_IN2_Y	(3ns,3ns)
tpd_IN3_Y	(3ns,3ns)
tipd_IN1	(0ps,0ps)
tipd_IN2	(0ps,0ps)
tipd_IN3	(0ps,0ps)

Tracing Timing Violation

- Keeping in mind all of the above information, we can now explain what causes the glitch. In the warning
- displayed in the **Console** window, the time of the glitch detection is 20040 ps (20.04 ns). This is specified in the line beginning with the Time clause:

```
KERNEL: WARNING: VitalGlitch: GLITCH Detected on port Y ;
Preempted Future Value := 1 @ 23 ns; Newly Scheduled Value
:= 0 @ 23.04 ns;
KERNEL: Time: 20040 ps, Iteration: 1, Instance: /AND3_0,
Process: VITALBehavior.
```

Tracing Timing Violation

- The next preempted value for the output Y port is '1' at 23 ns. However, the newly scheduled value for the output Y port is '0' at 23.04 ns.
- We should keep in mind that the current output value for the Y port is '0'. The absolute time period between those two transactions equals 40 ps and we know that the output delay for this particular gate is 3000 ps.
- If we add the present time value of 20040 ps and the output delay of 3000 ps then we will have the result of 23040 ps. This is the time of the newly scheduled value for the output Y port.

Tracing Timing Violation

- The design operates in the *OnEvent* mode…

```
CONSTANT DefGlitchMode : VitalGlitchKindType := OnEvent;
...
VitalPathDelay01 (...
Mode => DefGlitchMode,
```

- …where all input changes that have a duration time shorter than the output delay *tpd* are propagated to the output with 'X' values.
- The simulator, however, does not display the glitch on the waveform because the *Xon* generic has been assigned with *FALSE* value.

```
CONSTANT DefGlitchXOn : BOOLEAN := FALSE;
...
generic(...
XOn: Boolean := DefGlitchXOn;
...
VitalPathDelay01 (...
Xon => XOn,
```