How to Customize the ModelSim Wave View in the Altera Quartus Simulation Flow

Cristian Sisterna

Summary

When ModelSim is automatically lunched from the Quartus environment, it just displays the top level entity signals in the Wave View window. However, to either facilitate debugging tasks or check specific behavior of lower level components, most of the time internal signals also need to be displayed in the Wave View window panel of ModelSim. Moreover, coloring, ordering and grouping signals is especially useful when simulating complex designs. Therefore, having one or several custom views and invoking them automatically will help the verification job. This application note details the necessary steps to setting up the Quartus environment, as well as the ModelSim commands to open a customized Wave View window panel automatically.

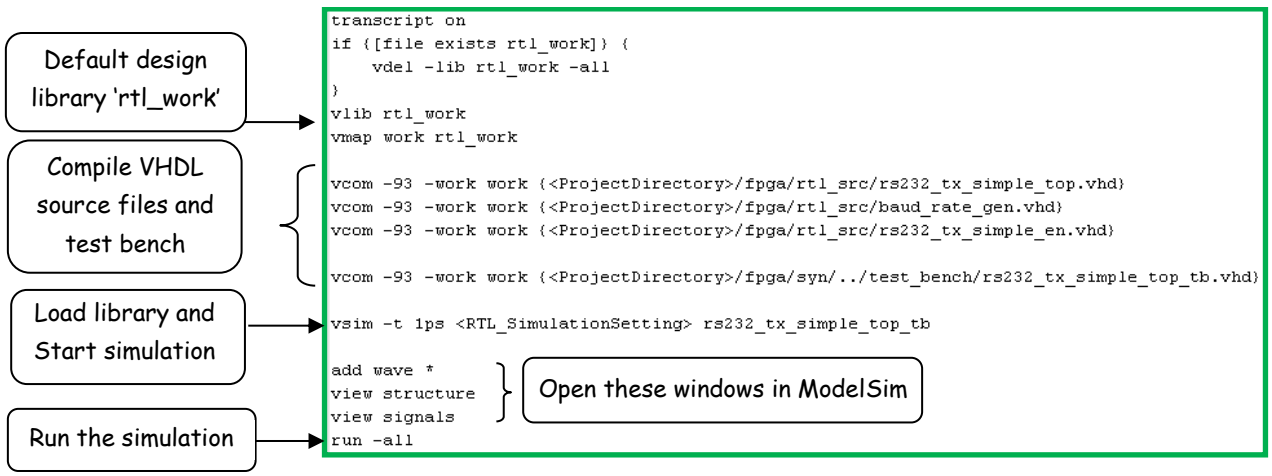
Scripts files (.do) created by Quartus

When ModelSim is launched from the Altera Quartus environment, Quartus automatically creates a `.do` script file that contains all of the necessary commands to compile the .vhd source files and the test bench, load, start and run the simulation in the ModelSim environment.

The `.do` script file created by Quartus has a name based on the type of simulation launched and on the name of the top level entity. For instance, for an RTL (Behavioral, Functional) Simulation the name of the script is as follow:

`<TopEntityName>_run_msim_rtl_vhdl.do`

A detail of the ModelSim commands found in a `<TopEntityName>_run_msim_rtl_vhdl.do` script file created by Quartus when running an RTL simulation is described below:



Likewise, when launching gate level simulation a similar script file is created. In this case the name of the script is:

<TopEntityName>_run_msim_gate_vhdl.do

The main difference between the gate level script and the behavioral script resides in the source files compiled when the scripts is executed. In the case of gate level simulation the <TopEntityName>.vho file and the test bench file are compiled. Whereas in the behavioral simulation, all the .vhd files that are part of the project are compiled along with the test bench file. On the other hand, there are some similarities between the scripts, the last four commands of both scripts, (see the figure above for the RTL script) are exactly the same. Among these commands the "add wave *" command orders to display all the top level entity signals in the ModelSim Wave window panel. However, Quartus offers to the designer the option of changing part of the automatically generated scripts to have a customized view of the simulated system. Following, the necessary steps to customize the Wave View window panel in ModelSim are detailed.

Waveform Customization Process

The different steps to follow to be able to customize the Wave View window panel in ModelSim when running the simulation flow in the Altera Quartus environment are:

1. Create a new Quartus project and write the necessary VHDL code or just open an existing project.

2. If the test bench has already been written go to step 3, otherwise write the respective test bench and add the file to the project.

3. Execute either the RTL Simulation or the Gate Level Simulation of the design:
    a. Tools -> Run EDA Simulation Tools -> EDA RTL Simulation
    b. Tools -> Run EDA Simulation Tools -> EDA Gate Level Simulation

    For instance, when executing the RTL Simulations a script similar to the one showed above is generated and executed. Thus, ModelSim (the simulation tool by default) is invoked and will come up to run the simulation as defined in the test bench and will open the window panels that are detailed in the RTL Simulation script just created, by default they are: Wave Window, Structure and Signals.

4. After the simulation stops, keep ModelSim open. The following steps will detail the customization process of the Wave View window panel.

5. ModelSim offers several techniques to customize the view in the Wave View window panel. In this step the most commonly used techniques will be described and detailed to later add them to the customized script file:
    a. *Adding internal signals:* to add internal signals to the Wave View window goes to the Instance window panel of ModelSim (left most panel) and find the test bench name. Underneath the test bench find the component instantiation label of the entity under test. This label is the label used in the component instantiation of the top level entity in the test bench code. It usually is something like UUT, U1, DUT or something similar. Click over the '+' symbol on the left of the

instantiation label to expand the component instantiations and be able to see all the components of the top level entity. Then, single click over the component (design unit) that holds the signal(s) you want to add to the Wave View window panel. Hence, the name of all the signals of the selected design unit will be displayed on the Objects window panel (usually the window in the middle of ModelSim). In case the Objects window panel is not visible, go to the bar menu and select View -> Objects. Then, single click on the signal to add to the Wave View window panel and then drag and drop the signal into the Wave View window panel. You can also do a single click over the signal, or keep Ctrl pressed and click on all the signals you want to add, then right click and select Add->ToWave->Selected Signals. Repeat this process and add as many signals as needed. Once a signal is added to the Wave View window, you can move it up or down as explained below.

b. *Moving signals*: it is also possible to move either up or down the signals displayed in the Wave View window to facilitate reading the waveforms. For instance, you can place all the input signals above of all the other signals, then the control signals in the middle part of the waveforms and the output signals at the bottom. To move up or down the signals in the Wave View window panel, select the signal you want to move (single left-mouse click over the signal). Then, drag the signal up or down by keeping the left-mouse button pressed and moving the mouse up or down, and drop the signal, wherever you like, by releasing the mouse button. You can select a group of signals to move up or down by using the Shift or Control keys as you usually use them.

c. *Adding dividers*: signal dividers can be added in the Wave View window panel to label the signals grouped with a specific purpose. For instance, you can add dividers labeled Input Signals, Control Signals, Rx Signals, Memory Read, Clocks and such. To add a divider select the signal over which you want the divider. Then, click the right-mouse button and select 'Insert Divider' from the pull down menu. The divider dialog window will come up. You can add a name describing the function of the signals that will be under the divider. You can also set the Divider Height, though this is not usually changed.

d. *Coloring the waveform*: for some specific purposes, for instance for a quick waveform reference among the myriad of waveforms, it is advantageous to change the color of a specific waveform as well as the color of the respective signal name. To change the color, first highlight (select) the name of the signal that you want to change the color, then right-mouse click and select Properties. The Properties window should come up. In the View tab, you will find the Wave Color and Name Color options. To change the default color of either the Wave Color or the Name Color, click on the respective Colors button and select the desired color. When finishing, click on Apply.

6. Once you are done adding signal, dividers and changing colors, it is then necessary to save this new customized Wave View window panel format to be able to use it again:
   a. Be sure to select the Wave View window panel among the other ModelSim views.
   b. Go to File -> Save Format. The Save Format window will come up. The default name for the wave format is `wave.do`. In case you are planning to have a customized wave view for the different kind of simulations (like RTL Simulation, Gate Level Simulation) change the default `wave.do` to a name something like `wave_f.do` for functional, or `wave_gl.do` for gate level. Then, click OK in

the Save Format dialog window to save the new wave format in the default directory or in the directory you would like to use by browsing to it.

7. Close ModelSim and go back to the Quartus main window.

8. Once the new format of the Wave View window panel has been saved either as `wave.do` or as your customized `.do` file, it is necessary to create a small script file that will hold the customized wave format , the time for the simulation to run and any other command that you want to add.

    a. In Quartus, go to File -> New -> Other Files -> Text File

    b. In the text file add the following lines:

        i. `do <your_wave>.do` (replace `<your_wave>.do` for the name used in point 6.b)

        ii. Add any other ModelSim command if it is needed (more on this at the end of this paper).

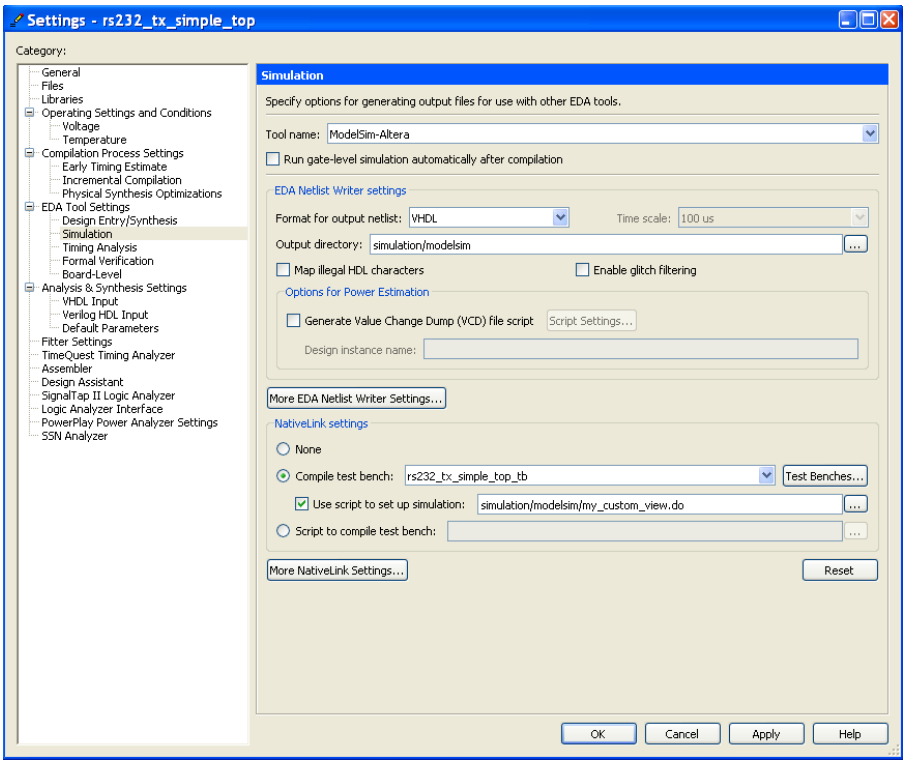
        iii. Add the simulation time. You can either write `run -all` or `run <time>;` i.e. `run 20 ms.`

    c. Save this text file in the directory: …/syn/simulation/modelsim/. with the extension `.do`. For instance you can save it as: `my_custom_view.do`

    For instance a simple `my_custom_view.do` file could have only two lines:

    - `do wave.do`
    - `run 20 ms`

9. As following step, it is necessary to instruct Quartus to use the customized Wave View panel format.
    a. Go to Assignments and select Settings. The Settings window will come up.
    b. From all the available options click on the EDA Tool Settings and then on Simulation. The different setting for the Simulation will be showed.
    c. In the 'NativeLink Settings' part of the window, bottom part, make a single click on 'Use script to setup simulation'. Then, using the browse button on the right, browse to the simulation directory or to the directory where you save the `.do` script generated in step 8. Select the respective script file, for instance, select the `my_custom_view.do` file. Hence, the Setting window should look like something similar to the figure below.



d. Click OK to save the settings and close the window.

10. From now on for every either RTL Simulation to be executed, ModelSim will use the custom waveform format that you configured. Exactly the same steps need to be followed in case you are customizing a Gate Level Simulation Wave View window panel. However, keep in mind that in case you have different customized views for the Gate Level and RTL simulations, it is necessary to have different names for the different scripts and change the script name of step 9.c for the required simulation.

11. As a final resume, the figure below shows the updated script with the custom script file that replace the original one showed in page 1.

```
transcript on
if {[file exists rtl_work]} {
    vdel -lib rtl_work -all
}
vlib rtl_work
vmap work rtl_work

vcom -93 -work work {<ProjectDirectory>/fpga/rtl_src/rs232_tx_simple_top.vhd}
vcom -93 -work work {<ProjectDirectory>/fpga/rtl_src/baud_rate_gen.vhd}
vcom -93 -work work {<ProjectDirectory>/fpga/rtl_src/rs232_tx_simple_en.vhd}

vcom -93 -work work {<ProjectDirectory>/fpga/syn/../test_bench/rs232_tx_simple_top_tb.vhd}

vsim -t 1ps {RTL_SimulationSettings} rs232_tx_simple_top_tb

do <ProjectDirectory>/fpga/syn/simulation/modelsim/my_custom_view.do
```

More on Customization

Besides of the `wave.do` and the `run` commands, other ModelSim commands can also be added to the script file with other purposes. For instance, one useful command for designs with a large amount of internal signals is the following:

```
log –r /*
```

This ModelSim command will log *all* the data objects in the design. This is very useful when, for instance, after running the simulation you find out that you would like to see an internal signal that is not currently displayed in the Wave View window panel. By using `log –r /*`, you just need to select the signal you want to add, drag and drop it in the Wave View. Then, its respective waveform will immediately be displayed. Without the `log` command, if that particular internal signal is not in the list of signals in the `wave.do` script, it is not logged; therefore, no waveform will be displayed for that signal. In such a case, it will be necessary to add the signal to the Wave View window, save the `wave.do` again and then rerun the simulation. The drawback of the `log` command is that it could make the simulation much slower since it needs to log *all* the signals of the design.

On the other hand, one point to keep in mind when dealing with the signals added to the script file(s) is the fact that the internal signals of the design after being placed and routed usually do not keep the same name as before the place and route. This means that the `wave.do` saved for functional simulation, might not be able to display the internal signals added to the script when doing a post-place and route simulation, that is the Gate Level simulation. A solution to this problem is to create one customized `wave.do` for the RTL simulation and another one for the Gate Level simulation using different names for each.

Another helpful point to be familiar with is the fact that it is possible to rerun the simulation without closing ModelSim, avoiding returning to Quartus. For instance, if you find an error when running the simulation in one of the .vhd source files, you can edit the VHDL file, still keeping open ModelSim (even you can use ModelSim text editor), save the modifications of the VHDL file and rerun the simulation by typing do <TopEntityName>_run_msim_rtl_vhdl.do or <TopEntityName>_run_msim_gate_vhdl.do and pressing enter in the transcript window (bottom window) of ModelSim. You can also use the up-arrow key in the transcript window of ModelSim to go through all the executed commands until you find one of the commands detailed above and then just press enter to execute it.