Tutorial

on

Simulation Using ModelSim

Introduction

Simulation is the process of applying stimulus or inputs that mimic actual data to the design and observing the output. It is used to verify that the design performs as expected and performs required functions.

Simulation can be performed in three places in a project design flow: after coding, after synthesis, and after implementation. The respective simulations are called functional, post-synthesis, and timing simulation, respectively.

Tools needed for simulation are: Text Editor and a Simulator.

You can use the built-in source editor in ModelSim.

ModelSim is a high-performance digital simulator for VHDL, Verilog, and mixed-language designs. It can be used for both FPGA & ASIC designs.

Students can use ModelSim for:

1. Functional Simulation of VHDL or Verilog source codes.
2. Post-Synthesis simulation of the circuit netlist.
3. Timing Simulation of the design obtained after placing and routing.

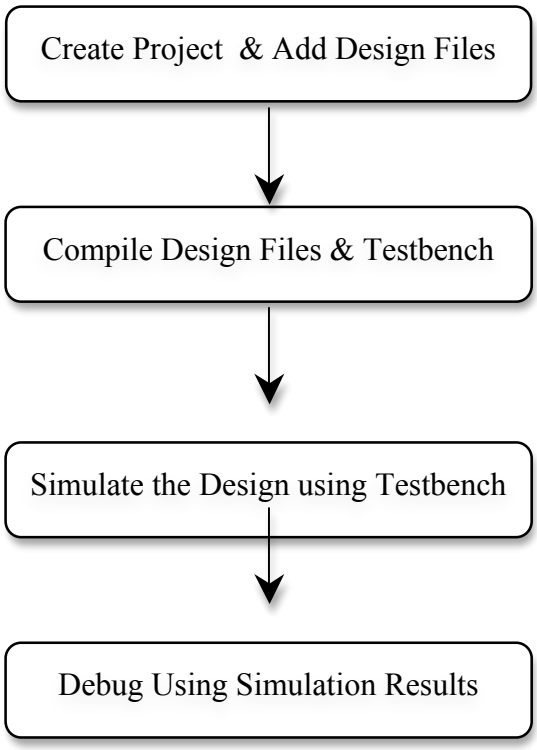
Objective

The tutorial will help you to:

1. Create a project and add your design files to this project.
2. Compile and perform simulation.
3. Debug source files.

Note: This tutorial does not explain the design flow, i.e. synthesis or implementation.

Simulation Flow:



(1) Create Project & Add Files.

In ModelSim, all designs are compiled into a library. You start a new simulation in ModelSim by creating a working library called "work". You select a destination for your project and give it a name.

(2) Compiling Design Files and Testbench.

After creating a project and adding files to it, you compile your design units into it. You can simulate your design if there are no errors.

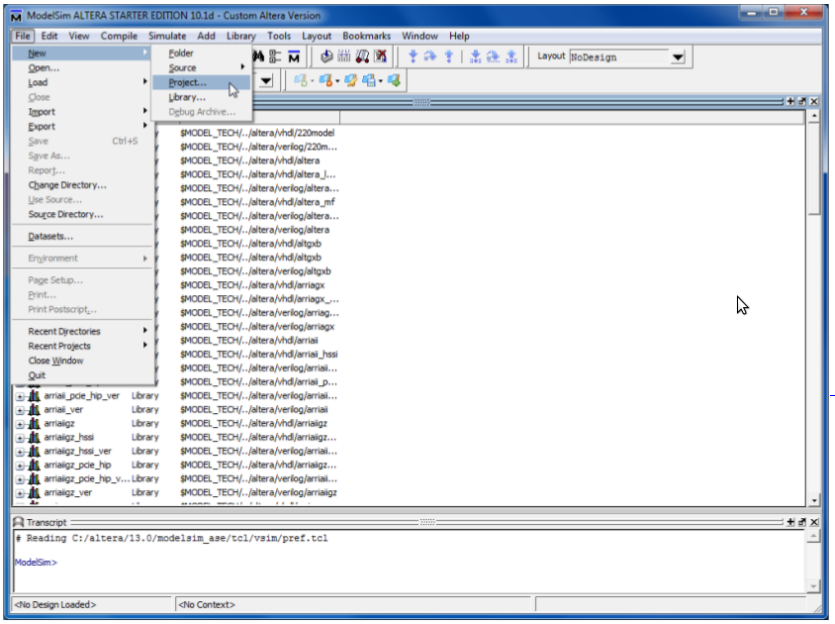
(3) Simulate the Design using Testbench.

With the design compiled, you invoke the simulator on a top-level module (which is the Testbench, as you have instantiated your top level design entity in it). Assuming the design loads successfully, the simulation will run and you will see timing waveforms.

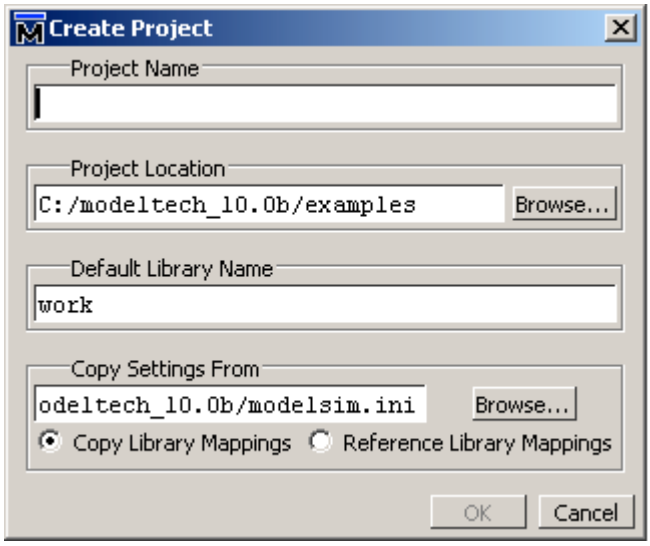
(4) Debug Using Simulation Results

If the results are not as you expected, you could use debugging environment to track down the cause of the problem.

(1) Create Project & Add Files.



Click on "**File**" → "**New Project**". In the "**Project Location**" point it to where your folder has been created.



When the next window comes up as shown in the next page, click on Add Existing File, and browse to the folder where the files are stored or click on Create New File and select file type as verilog. The will be added to the project space.

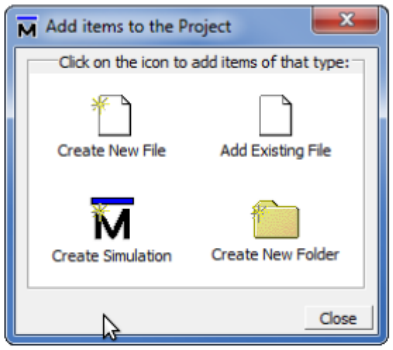


Figure 3. Add Items window.

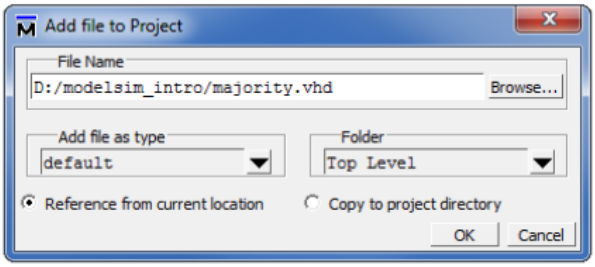
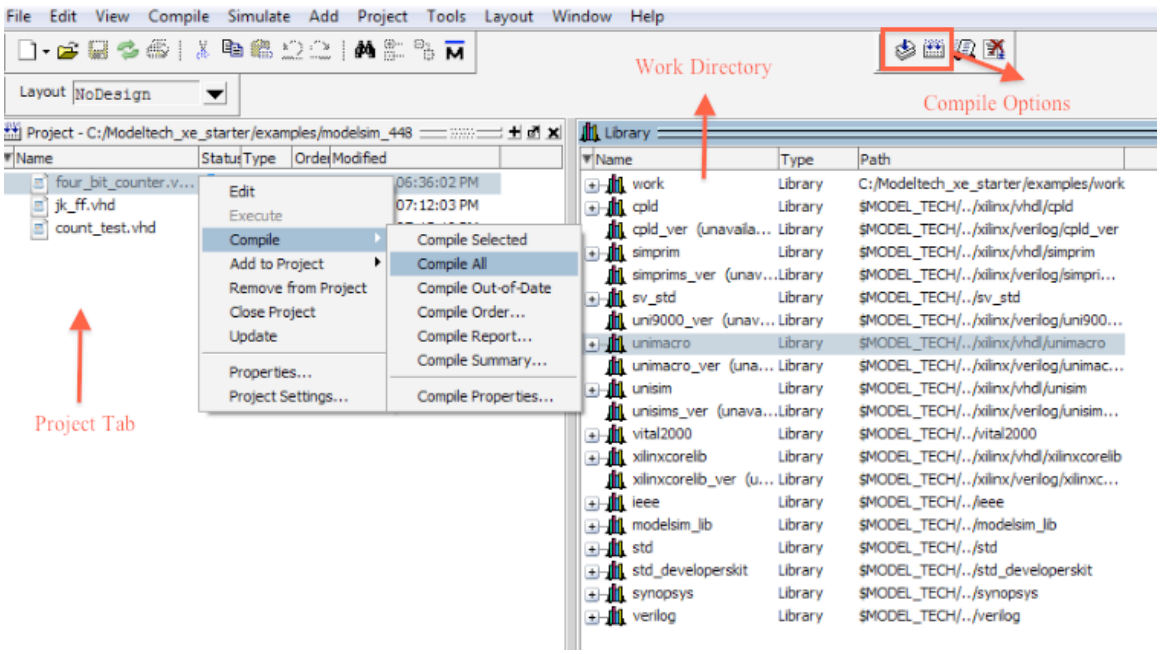


Figure 4. Add Items window.

(2) Compiling Design Files and Testbench.

Right click on any file and select **Compile→Compile all**.



The other option is to choose **Compile** from the **Menu** and click on **Compile all.**

You can also choose one of the following icons  . The first icon compiles the selected file, the second compiles all.

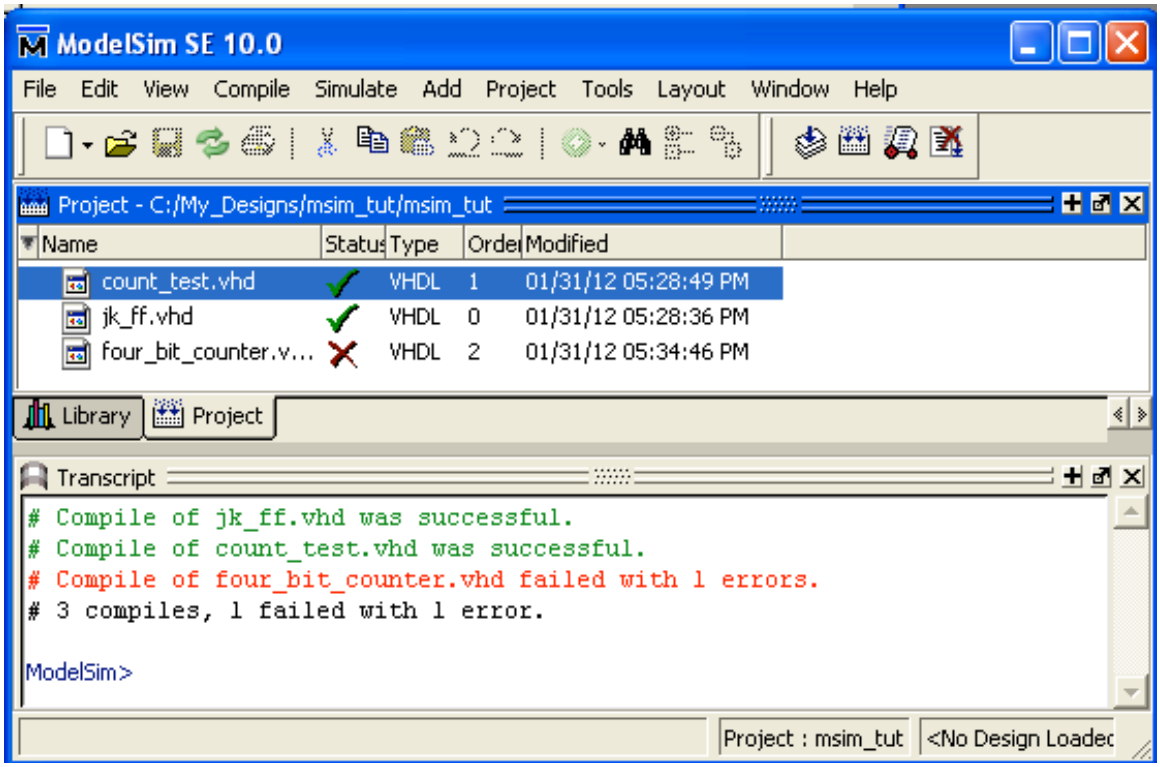
If everything goes well, you will get a **Green tick** next to your files. To see what happens when there is an error, change line 7 of **four_bit_counter.vhd** from

"counter : OUT STD_LOGIC_VECTOR(3 DOWNTO 0)

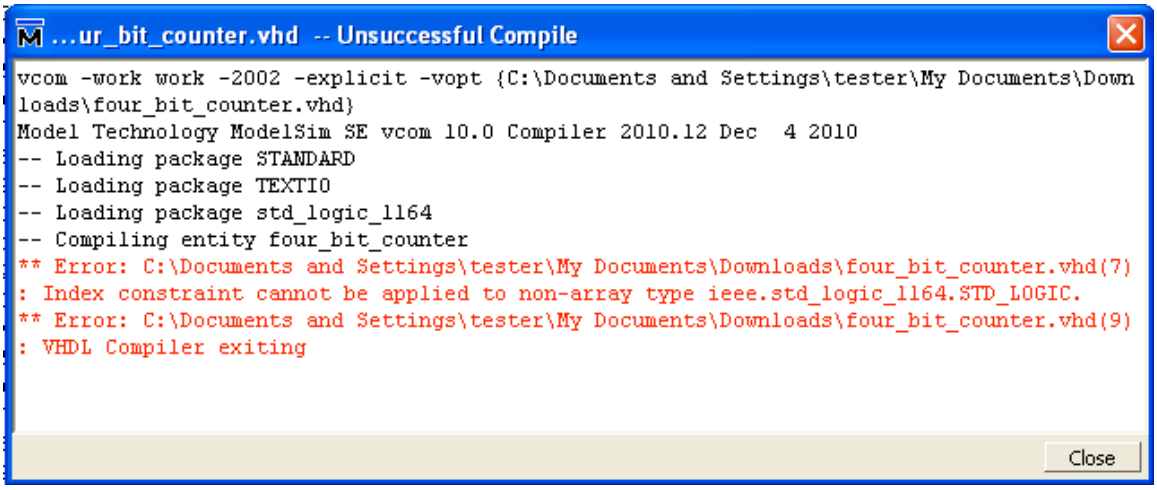
to

counter : OUT STD_LOGIC (3 DOWNTO 0)

Recompile this file. There will now be an error message in the message window, as shown below, and you will see a **Red Cross(X)**.



To find out what this error is and in which line it originates, **double click** on the **error**, and a window will pop up with more description about the error as shown below.

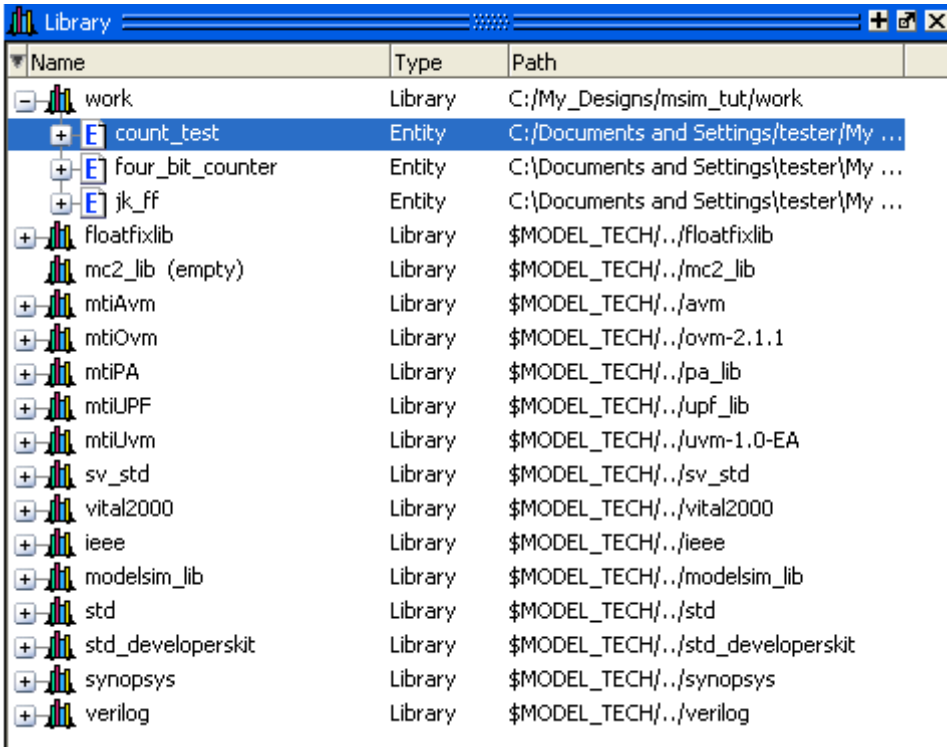


If you **double click** on the **error description**, a small editor window will open and the error will be **highlighted**. After correcting and **saving** the file **compile** again.

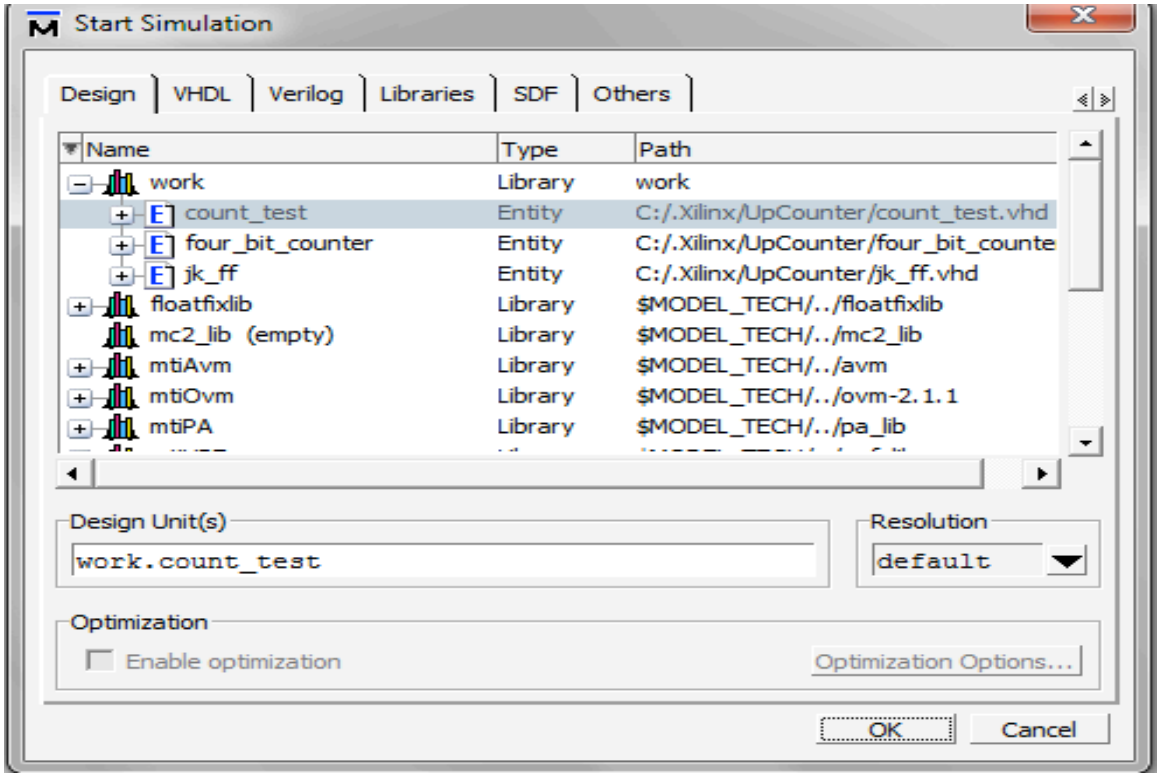
```
ENTITY four_bit_counter  IS
PORT (     clk:      IN STD_LOGIC;
           reset:       IN STD_LOGIC;
           counter : OUT STD_LOGIC(3 DOWNTO 0)
     );
END four_bit_counter;
```

(3) Simulate the Design using Testbench.

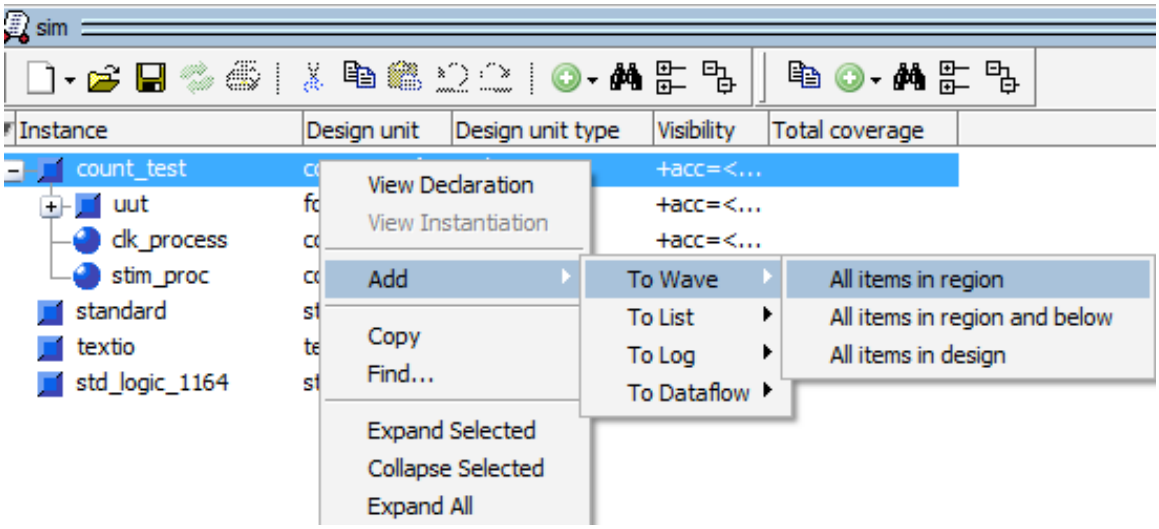
There are two ways to load your design and simulate it. The easier way is to **expand** the **work library** by clicking on the **plus** sign. The work library is in the **library tab** under **workspace**. If you do not see the library tab, go to the "**View**" menu and make sure that **"Library"** is selected. Double click on the **testbench** (count_test.vhd) and the design will be loaded.



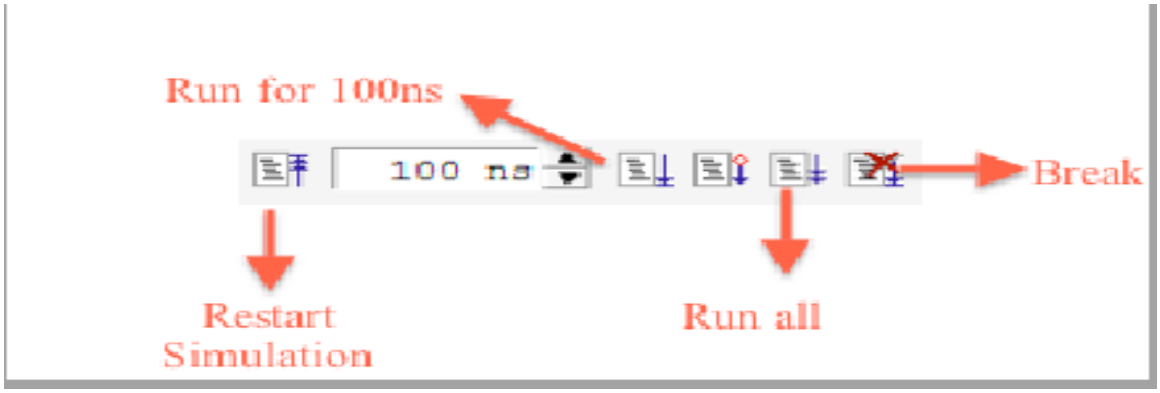
The other way is by clicking or  (simulation). It is an icon right next to the compile icon. A new window will pop up. Select the testbench and click OK.



This will load your design and a new tab will open called **SIM.** It will contain your testbench and the design under test, which is instantiated in it.

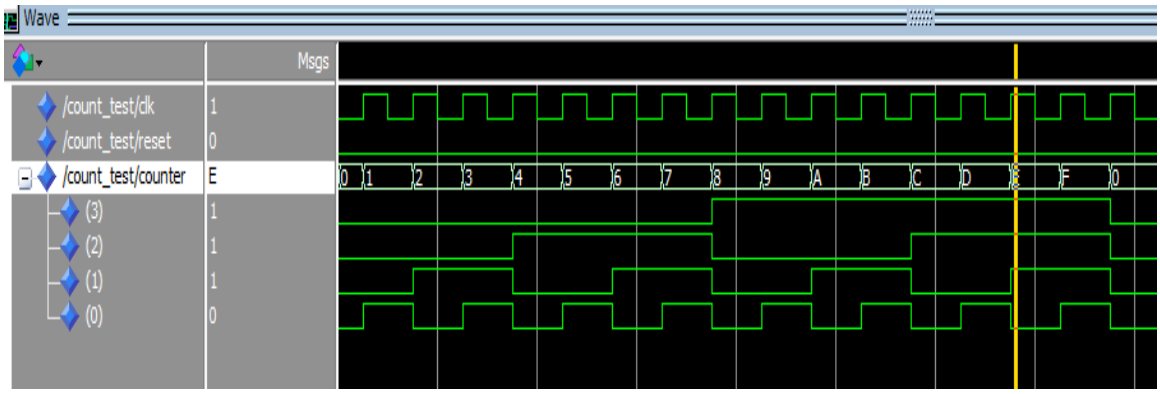


Right click on the **testbench entity** and **Add it to the wave** as shown in the window above. Now switch to the **wave window**. If you do not see the wave window, go to "**View**" → "**Wave.**" You should now see the added signals in the waveform window. The next step is to run the simulation. There are several run commands on the toolbar.



The **Run for** icon  , will run the simulation for the amount of time specified in the time field. In this example, it is 100 ns. **Run-all** will stop when there is a **wait statement** in the testbench or when you hit break.

The waveform will not look like this when the simulation ends:



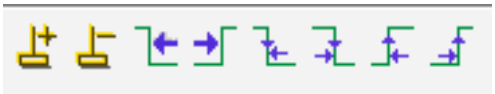
You will need to zoom in and this will be discussed in the next topic.

(4) Debugging your results

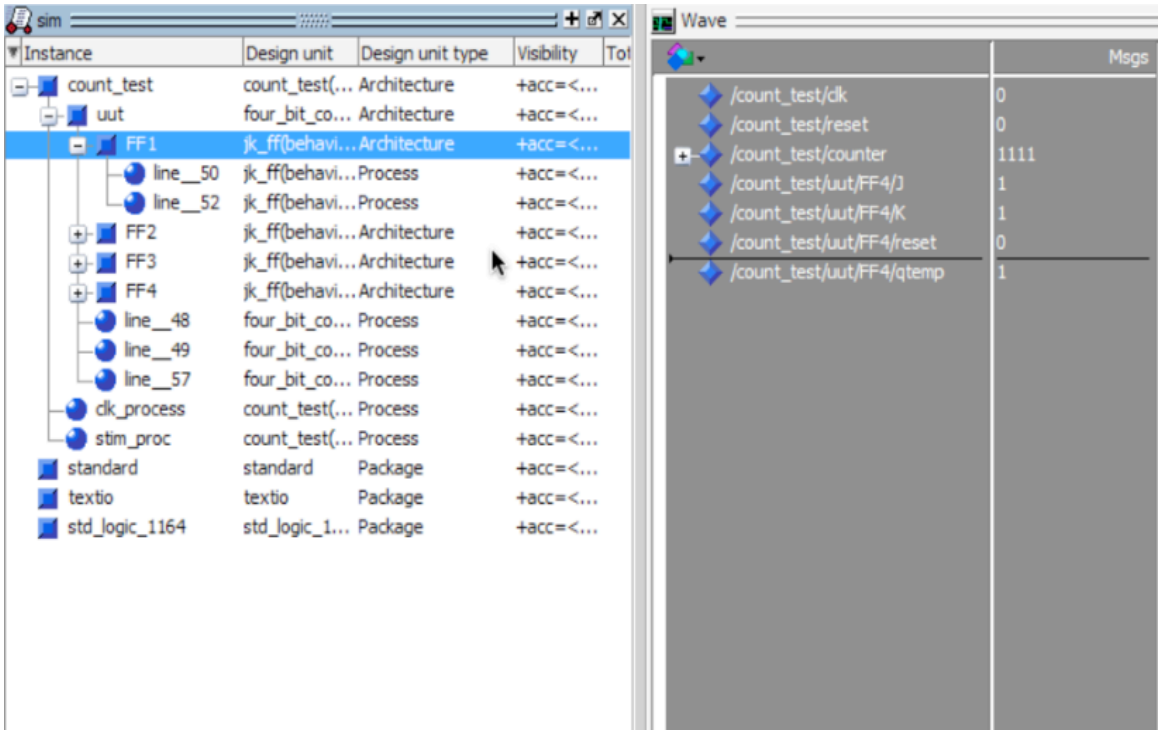
After the simulation ends, you will need to fit the waveform using the zoom icons shown below.

The first two icons are for zooming in and zooming out respectively. The third icon means that the waveforms should fit the screen. The last icon zooms in at the interval next to the position of the current active cursor.

To help with debugging there are options such as adding cursors and finding the next transition. Starting from the left, the **yellow +** adds a cursor line, and **yellow -** deletes a cursor line. The next two icons will help you find the next transition on a selected signal referenced by the cursor. The next four are for finding the next or previous, falling or rising edge of the clock.



A very important thing to know is to be able to add internal signals from your top-level entity to the waveform. It is very helpful in debugging, when you are not getting the correct/expected output. Click on the **instantiated module**, which is under the testbench entity in the **SIM** window. Right click on any smaller components and **Add to wave.**



We added the signals in **FF4**, which is the fourth JK flip flop in the circuit. To see waveforms for these new signals, you need to **restart** the simulation and **run again.**

You can change the **color** of a **signal wave** by **right clicking** on the signal and selecting properties. You can set **color** as well as **radix**. You can also **group** related signals together as well as add **dividers** to make your waveforms more readable. There are many options and you should do your best to explore them.

The above icons can be used to **Debug** the code very effectively. These icons help by making you go through each step of your code. Starting from the left, the first icon is a **basic step debug**, second will **skip** to the **next step,** and the third will allow you to come out of the **loop/process**.

Saving the window format

If you close the Wave window, any changes you made to the window (e.g., signals added, color set, etc.) are discarded. You can use the **Save Format** command to capture the current Wave window display and signal preferences in

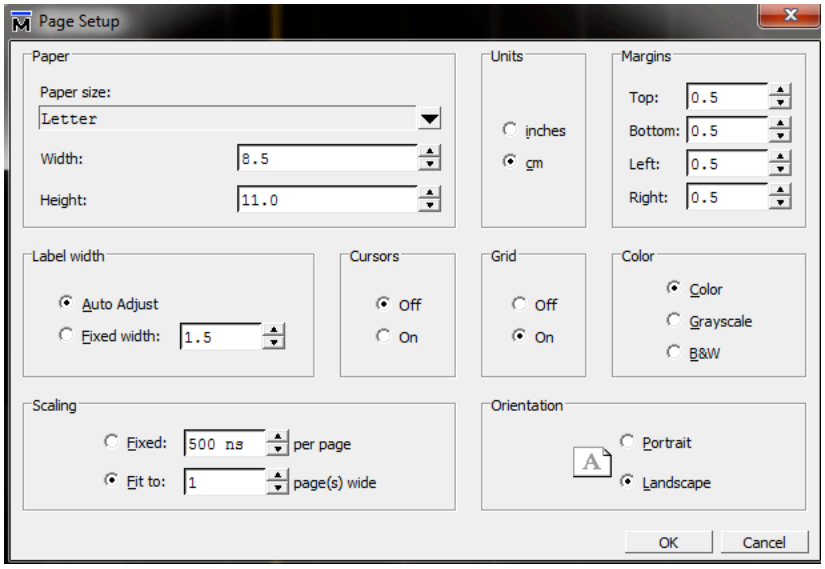
a DO file. You can open the DO file later to recreate the Wave window.

1 Save a format file.
   a. Select **File > Save > Format**.
   b. Set file name to *<name_you_like>.do* and click **Save**.

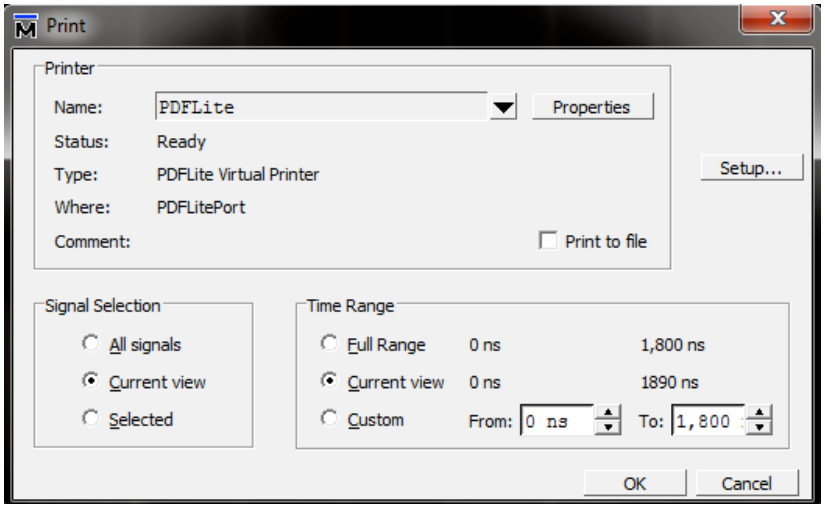
2 Load a format file.
   a. In the Wave window, select **File > Open > Format**.
   b. Select *<name_you_like>.do* and click **Open**.
   ModelSim restores the window to its previous state.

Taking a print out and saving the waveform in a PDF file

Under File menu, click on "Page Setup" option. It is very important to specify whether you want the output on single or multiple pages. Select the number of pages in such a way that waveform is clearly visible and information is not too cramped or spaced apart.



Now click on "Print" option. You can either choose full range or specify the specific range of simulation.



Take a print out by choosing any 3rd party tool (PDFLite, PDFCreator etc) to create a PDF file of the waveform. Here is a sample of the waveform output saved in the PDF file.