

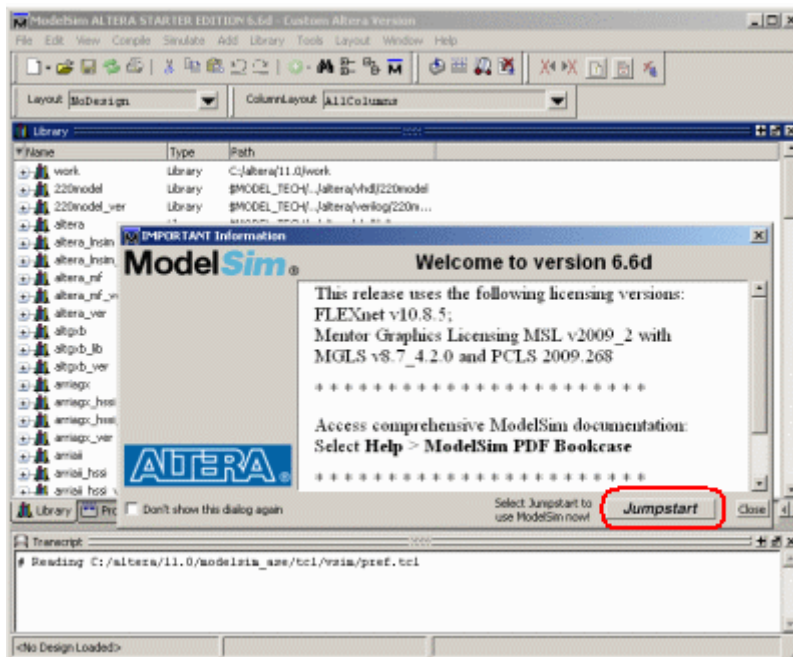
# Testbench in ModelSim



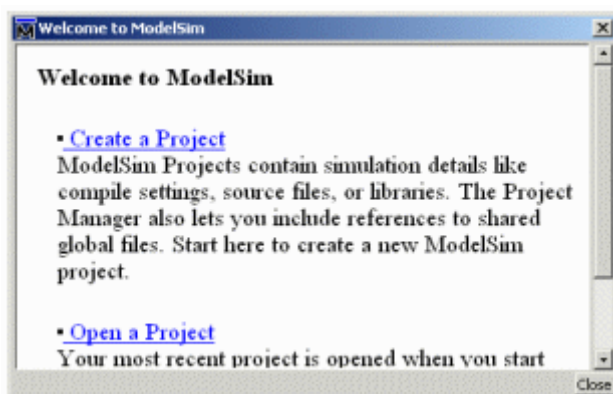
*Select the correct software version - in school there are several versions installed in the Start menu!*

```
Altera 13.0.1.232 Web edition\  
  ModelSim-Altera Starter Edition 13.0.1.232\  
    ModelSim-Altera 10.1d(Quartus II 13.0sp1)
```

Start ModelSim.

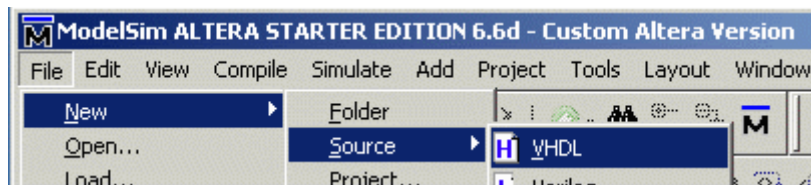


Klicka på **Jumpstart**. This time we choose "Open Project" for continue with our previous MAXsim-project.




## Testbench

In addition to the VHDL code for the lock, we now need another VHDL file for the test bench code.

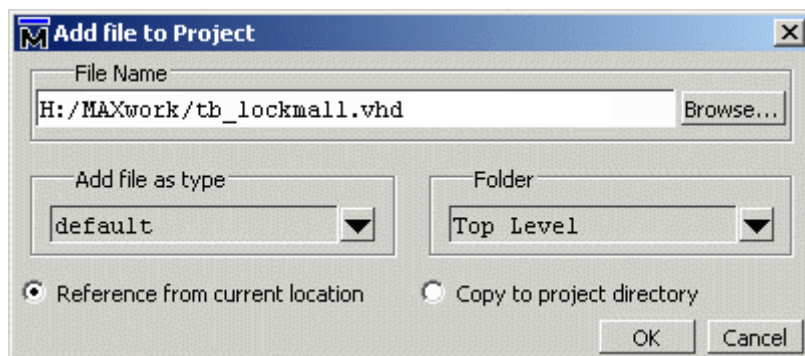
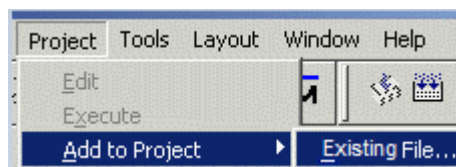


Create a new empty VHDL-file. Copy and paste the content of the file `tb_lockmall.vhd` and then save the file using the same name, `tb_lockmall.vhd`, among the other files in the project.





Alternatively, you can copy the file `tb_lockmall.vhd` to the folder with the other files of the project.

 [tb\\_lockmall.vhd](#) ([tb\\_lockmall.vhd.txt](#))





### Add the VHDL-file to the project



The project now has two files. The file `tb_lockmall.vhd` is not compiled yet, which can be seen on the blue question mark.

Project - H:/MAXwork/MAXsim			
Name	Status	Type	Order
 tb_lockmall.vhd		VHDL	1
 codelock.vhd		VHDL	0

Choose the menu **Compile** and the alternative **Compile All**. Now the file is compiled `tb_lockmall.vhd`. By the content of the file we can see that it controls the other file `codelock.vhd`, so that's why it gets the highest order.

Name	Status	Type	Order
 tb_lockmall.vhd		VHDL	1
 codelock.vhd		VHDL	0

## Load the Design to the simulator.

We want to test the bench messages to appear (as arrows) at the top of the **Wave**-window. Therefore, we write in the **Transcript** window:

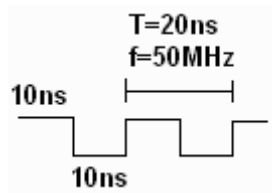
```
vsim -msgmode both -displaymsgmode both tb_codelock
```

Then add the signals in the **Wave**-window.

```
add wave *
```

## Start simulation

In the file `tb_codelock.vhd` you find `clk <= not clk after 10 ns;`. It creates the clock pulses with a period of 20 ns, as we used before.



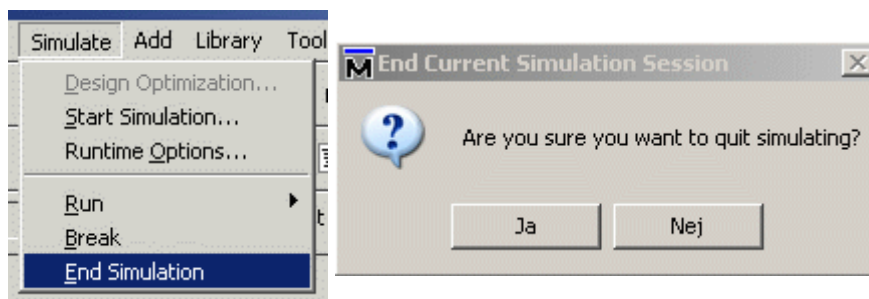
We can immediately start the simulation with the command `run 3us` in the **Transcript**-window. The time 3  $\mu$ s sufficient to try all possible key-press combinations.

In the **Transcript**-window we can read the encouraging message `Note: Lock tries to open for the right sequence!.`

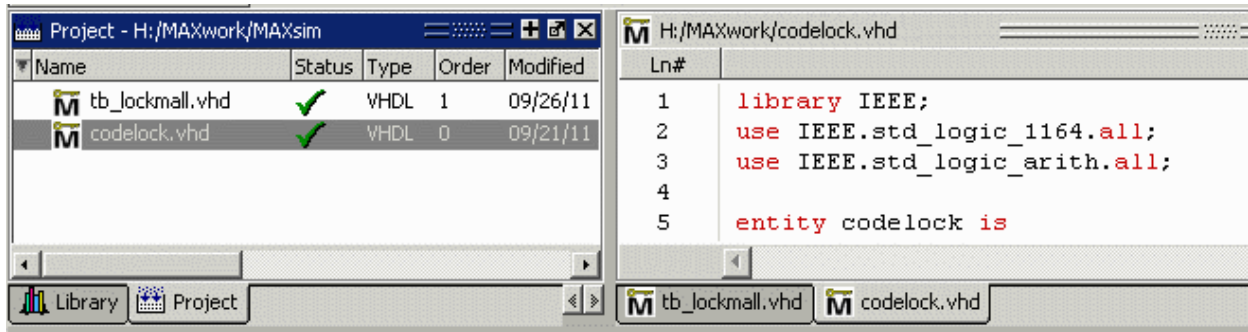
In **Wave**-window choose **View**, **Zoom**, and **Zoom Full** and then you can see the entire process.




The green arrows show the *when* the desired event happens!  
Now exit the simulation.



## Reveal the wrong code!



Now we need to in **Project** window, double-click the file `codelock.vhd` so that it appears in the text editor where we can change it. Copy and paste the content of the file `lockmall_with_error.vhd`. Then save the file under the same name as before and recompile everything.

 [lockmall\\_with\\_error.vhd](#) ([lockmall\\_with\\_error.vhd.txt](#))

Simulate in the same way as before. In **Transcript**-window you can read the message from the test bench when event happens!

The red arrow in the **Wave**-window shows *when/where* the undesired event happens!



## VHDL-test bench file

You do not of course be able to write a VHDL Test Bench file after a short first course on Digital Design. Still take the opportunity to go through the file and see if you can understand how it is intended!

Se **VHDL testbench file**

William Sandqvist [william@kth.se](mailto:william@kth.se)