# Laboratory-3: VHDL Simulation

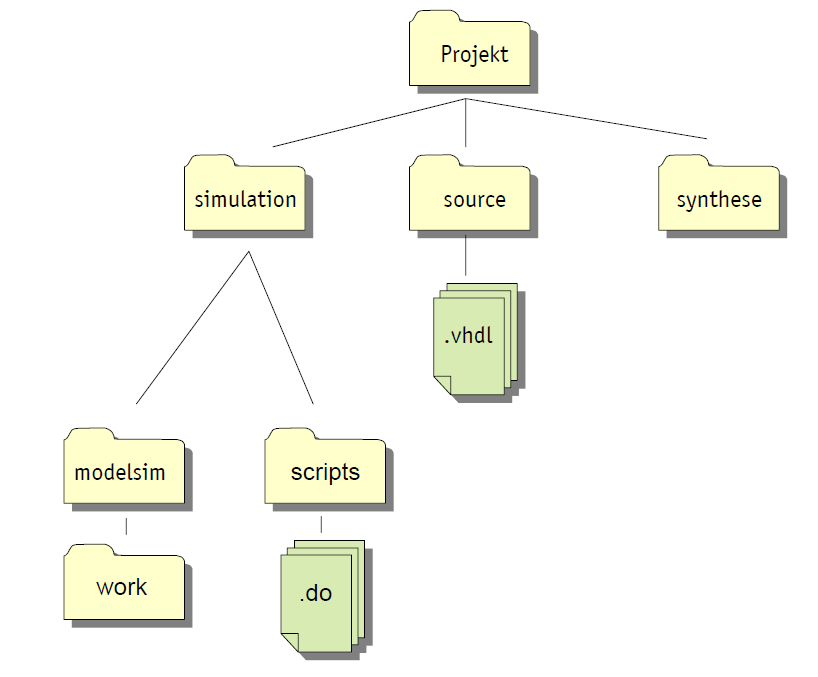
**What you should know after this laboratory:**

1. Create a ModelSim project to simulate a VHDL testbench, using a do-file script
2. Adapt and develop a simple testbench to test the functionality of an RTL design
3. Track the simulation execution in the waveform window and run-time messages (transcript window)

## Testbench example project

Take the files presented in the theory lesson as testbench example, and organize them in a new directory structure:

* **Lab3\_proj /**
  + **Source**: all \*.vhd
  + **Simulation** / Modelsim : here you create the ModelSim project
  + **Simulation** / Scripts: all \*.do (e.g. compile.do / wave.do )
  + **Synthesis**: here you create the Quartus project



Open ModelSim-Altera as standalone tool, which is found under:

<alteral\_installation\_directory>/<version\_number>/modelsim\_ase/win32aloem/modelsim.exe

Eventually there is already a shortcut defined in the desktop for the PCs in the lab.

Create a new project in the location indicated above, and include all VHDL source files you require (RTL blocks and testbench).

Start the simulation by calling the execution of the compile.do script, in the command line (in the bottom of the transcript window).

* do ../scripts/compile.do

Since you do not have yet a wave.do file, the script will crash with an error message. Select in the Objects window the signals you want to visualise, add them to the waveform and save a wave.do file in the directory indicated above. Observe that you can add signals on the top hierarchy, but also within your RTL block.

Restart the simulation (repeat the do command above), and analyse the waveforms generated:

* Do the signals behave as you expect?
* Which tests passed or not?
* How and when did the simulation stop?

## Creating a VHDL Description for the UART-RX Top

We want now to simulate your UART-RX project from Lab-2 with a VHDL testbench.

In order to do that we need first a VHDL description of the UART-RX top level schematics. Quartus can generate this hierarchical VHDL file, but you should first check/improve some points in your BDF schematics:

1. Give names to all signals (these are the connections inside your top level schematic):

select wire/right mouse click / properties

1. Make sure that you did not use the same name for a signal and a component (not supported).
2. Call "File / Create\_Update / Create HDL Design File from Current File ..." and select as option "VHDL" for the generated code.

The prerequisite is that there are no further circuit diagrams or Quartus schematic library components below their top level circuit diagram. If this is the case, you must replace it with VHDL code.

1. Name this top level VHDL code uart\_rx\_only\_top.vhd. Open uart\_rx\_only\_top.vhd and structure the generated code with comments.

Try to understand the entire code. Does the code meet your expectations? If there are still any connections with the name "SYNTHESIZED\_WIRE\_XX" in the code, you can return to the schematics, add the missing names and repeat the process in c) again.

1. Now remove the previous toplevel circuit diagram (bdf file) and insert the new file uart\_rx\_only\_top.vhd into the project. Synthesize the project in Quartus and check for the warning messages.

## Simulating the Testbench for the UART-RX Project

You have received a VHDL testbench to check your UART-RX implementation. Include this file in the source folder of your UART-RX project, then create a simulation folder and follow the steps described in section 1.1 in order to run a simulation of your UART-RX top level.

Show your simulation output to the teacher, and explain which tests are carried out in this testbench.