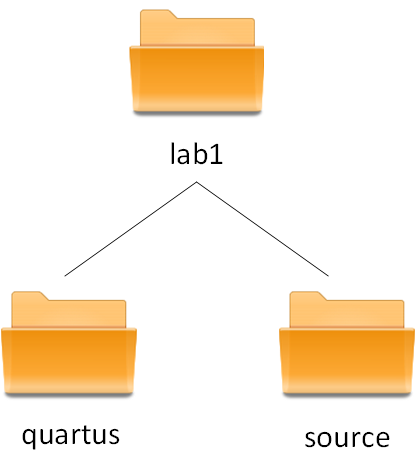
# Laboratory-1: Counter from Schematics to VHDL

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

1. Directory structure for a quartus project
2. How to describe simple counters in VHDL
3. Create a symbol from a VHDL file, and integrate it in a circuit schematics
4. Synthesize, download and test a circuit into the FPGA board

## Open an existing Project

The lab1 Quartus project is given to you as a start point. Observe the directory structure of the project, which is shown in the figure below.



**Figure 1.1:** Directory structure for a Quartus project

The files you write / design or import from another project ( source files \*.vhd, \*bdf, …) should be kept under the source directory. The files generated by Quartus toolset are found under the quartus directory (which means the quartus project was created here). Use this directory structure for future projects.

Open the project, open the schematics of the top level and analyse the circuit described here. Check your understanding by answering the following questions:

* How fast does the counter output change?
* What is the counter output range?
* Which output is shown in the seven segment display?

Import the pin assignment described in the csv file, compile the project, check the compilation results, download the sof file to the FPGA board, and check its behaviour.

## Describe a counter in VHDL

Replace the counter, which is currently described with schematics (full\_adder, flip-flops, plus constants) by a description in VHDL. In order to do that, create a new file, simple\_counter.vhd , then include it in the project, generate a symbol for it and adapt the schematics of the top level circuit accordingly.

## Counter Variations

Implement at least, one of the following counter variations (replacing your initial simple\_counter.vhd file):

* A modulo counter, which counts up or down, depending upon an external selector (via switch or button);
* A 4 bit counter covering the range 0-9. After 9 it overflows back to 0, and starts counting again;
* A 4 bit counter with inputs: load\_value (via switches), and start (via button). Once start is pressed, the counter takes the start value specified via switches, then counts down to 0 and stays there until a next start input (via button).

Present your results (code, compilation results, plus hardware checks) to the teacher.