# Laboratory-2: UART Receiver in VHDL

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

1. Create a new project in Quartus, using separated folder for source files
2. Analyse the requirements of a digital circuit, and translate into the required RTL structure
3. Decide the partitioning for the VHDL blocks. Design the blocks (VHDL) and top (schematic)
4. Test your overall functionality in hardware

## Study Requirements and Define Partitioning

Analyse the UART receiver (UART-RX) requirements, which were discussed in the theory lesson, and discuss with your laboratory partner to decide about the following design options:

* Include or not a clock divider, and adapt accordingly the count values for the baud tick generator
* Grouping FSM and counter, and/or shift-register in single block, or keep separated blocks
* Design a pure Moore FSM (outputs active during a state), or a Mealy FSM (outputs active during transitions between states)
* Stopping or not the tick generator once the reception of the data byte is done
* Decide to indicate (for example via LED) if an error is detected (wrong value of start- and stop-bit)
* Add parametric values (via generic) in your design blocks
* Decide the naming style for your signals and block-level ports.   
  Obs.: the port names for the top-level are fixed because of the FPGA pinning file

Draw a block diagram of your planned implementation, and give names to the blocks, ports and signals. Choose adequate naming, such that the action or functions related to these blocks and signals are easy to recognize.

Identify which blocks are new, and draw for the new blocks the RTL diagram. Do not forget to name your signals and indicate the width of the multi-bit ports and signals.

Show your block- and RTL-diagrams to the teacher, and explain your design options shortly, before you start the design of the new blocks.

## Implementation Steps

Follow the list below for the implementation of your UART receiver:

* Create a new Quartus project with the same directory structure as lab-1:
  + <proj-folder>/ source
  + <proj-folder>/ synthesis
* While creating your Quartus project, pay attention to:
  + Select the adequate FPGA device
  + Include the required design files ( <proj-folder>/ source/\*.vhd )
  + Create the project under the directory ( <proj-folder>/ synthesis )
* Design the missing VHDL blocks required to implement your UART receiver (UART-RX).
  + When you create the symbol for a VHDL block within Quartus, the block code is compiled and checked for syntax errors. Check if any errors and/or warning messages are generated, and correct them.
* Design the top-level schematics and compile the full project **before** reading the csv file for the pinning assignment. Check for errors and warnings (for example severe warning –violet ones-, and ordinary warning –yellow ones-). Which warnings do you judge that cannot be ignored/accepted?
* Read the pinning file and recompile the project. Open the programmer and download it to the FPGA board.
* Check the compilation reports, and find out the total number of flip-flops in your UART-RX. Check also under Analysis & Synthesis > Resource Utilization by Entity, and find out the number of flip-flops on your new VHDL files. It has to match your RTL diagram (from part 1.1).
* Go to Tools > Netlist Viewers > RTL Viewer and examine the RTL structured synthesized by Quartus. Specially check in detail your new blocks.

## Testing and Debugging your Implementation

Test your UART-RX by connecting your board with a jumper wire to the board on the front pult, which has an UART-TX.

In case your design is not working, try one or several of the following strategies:

* Double check the warnings of the synthesis (for example it should contain no latches, because these indicate incomplete combinational processes)
* Think about auxiliary signals, which you could wire to ports and observe them. In this case, you may need to reduce the clock frequency.
* Start a simulation with File> New> Verification/Debugging Files > University Program VWF
* In case you know already how to start a simulation with Modelsim, you can also do so. Otherwise use the other debug strategies and wait for the coming weeks, when we introduce testbench design in VHDL.

Once your UART-TX is ready and functional, demonstrate the working test to the teacher.