# **Step by Step Implementation**

This will provide you with a visual demonstration of the implementation process. If you are already familiar with Verilog synthesis and simulation platforms, you may skip this section. We will be using Quartus II (or Quartus Prime, if available) and ModelSim for this demonstration.

First, navigate to the extracted folder and open the 'uart\_tx\_rx.qpf' file.

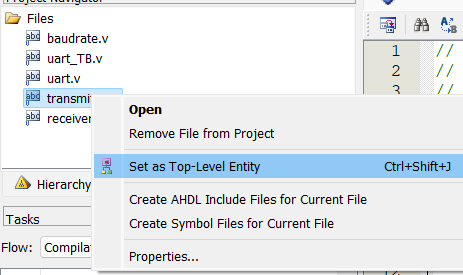


Next, we can access the code files by navigating to the 'Files' section.

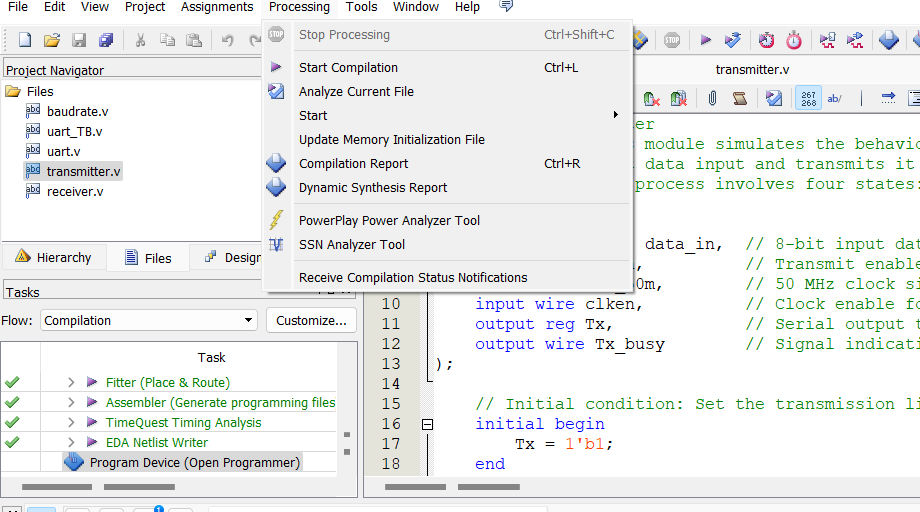
A screenshot of a computer

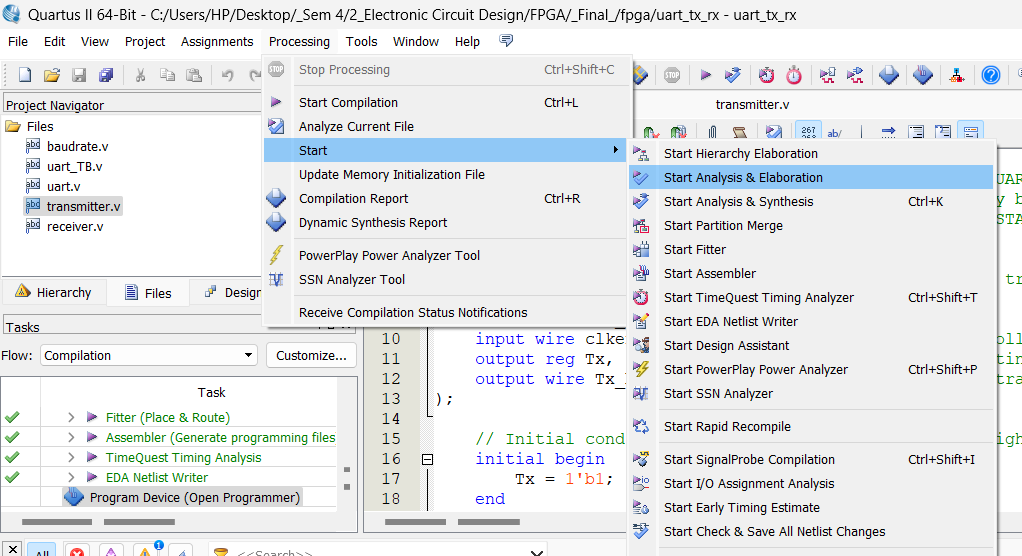
Description automatically generated

Here, I will demonstrate how to synthesize the transmitter module. You can follow a similar process for the receiver, baudrate, and uart files. However, please note that the 'uart\_TB' file should not be synthesized as it is meant for simulation purposes. Ensure that you set the top-level entity to the file you intend to synthesize.

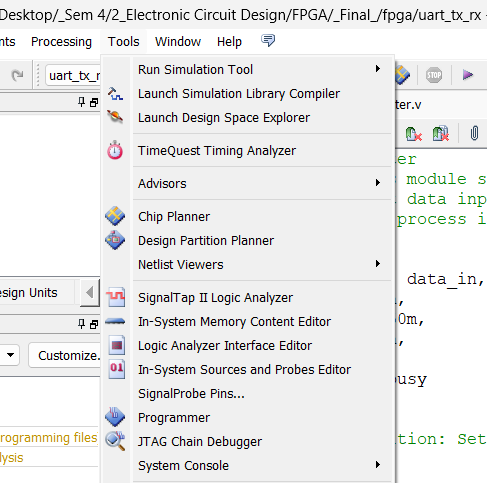


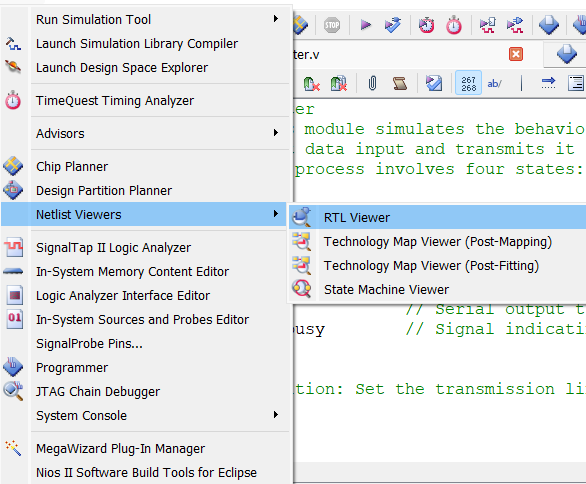
To synthesize the circuit virtually, navigate to Start >> Start Analysis & Elaboration.

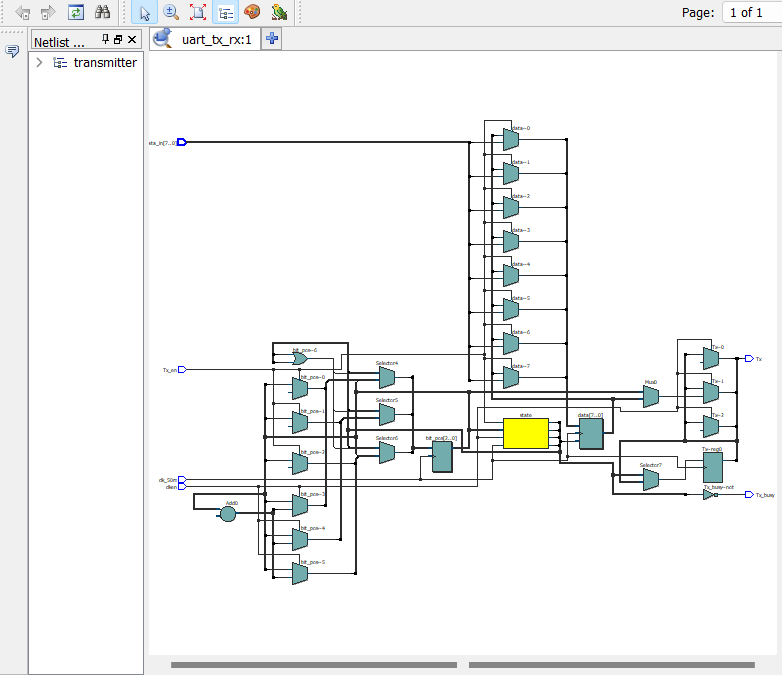




Then, go to Tools >> Netlist Viewers >> RTL Viewer to visualize the synthesized circuit for the transmitter module.

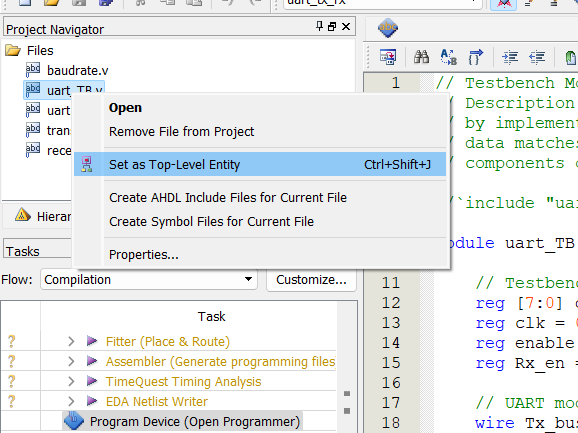


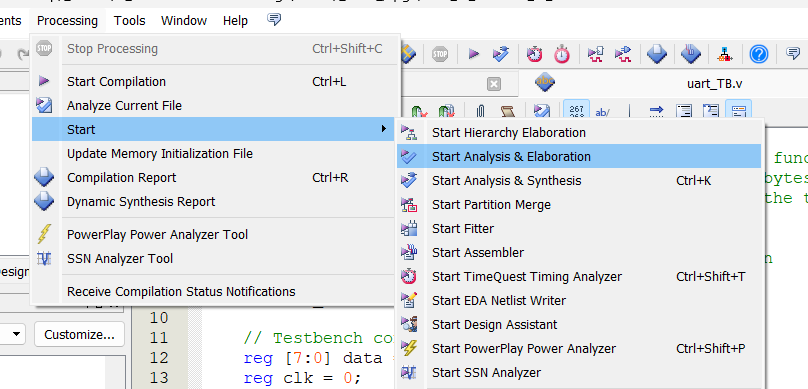




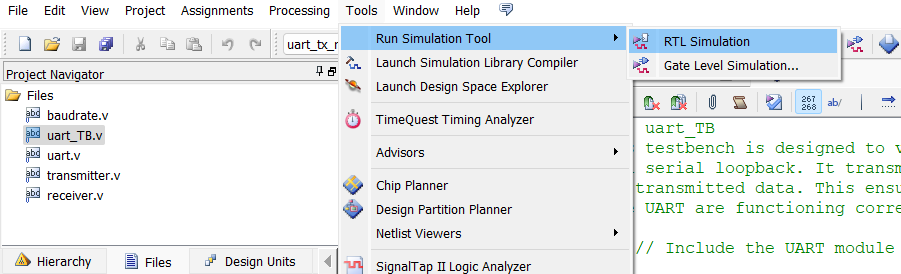
You can follow a similar procedure to synthesize other modules as well.

Now, let's proceed to simulate the testbench. The provided testbench is for the main uart circuit, but you can create testbenches for other circuits as well. Experiment with them to gain more understanding. Ensure that you have analyzed and elaborated all the circuits before simulating the testbench.

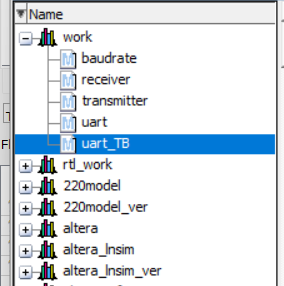
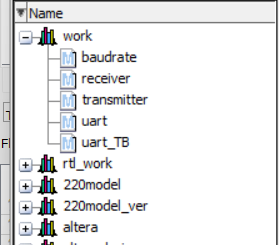




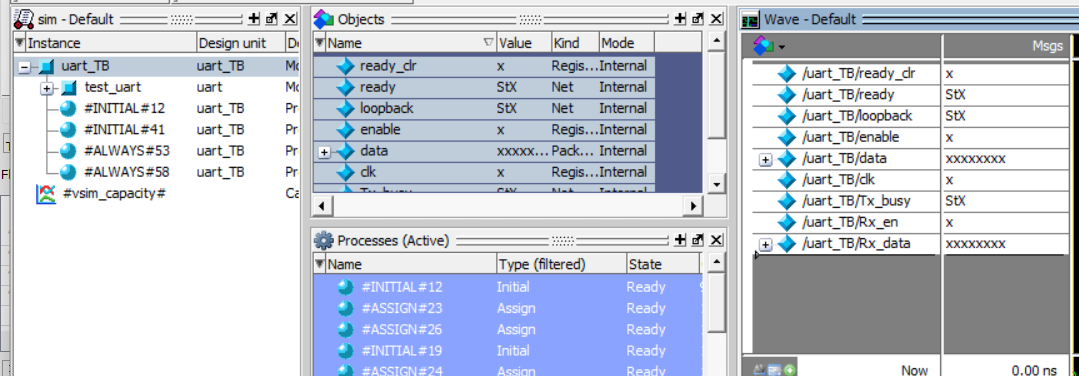
Now, let's proceed with the simulation.



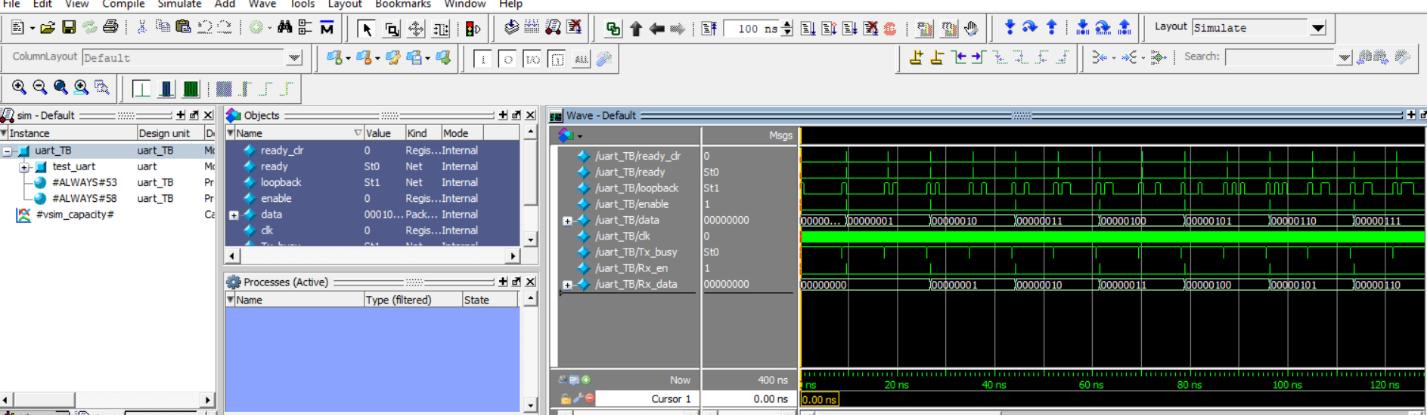
In ModelSim, navigate to the work directory and double-click on 'uart\_TB' (testbench).



Next, drag and drop all the entities from the Objects bar to the Wave bar. You can use the shortcut CTRL + A to select all entities and then drag and drop them.



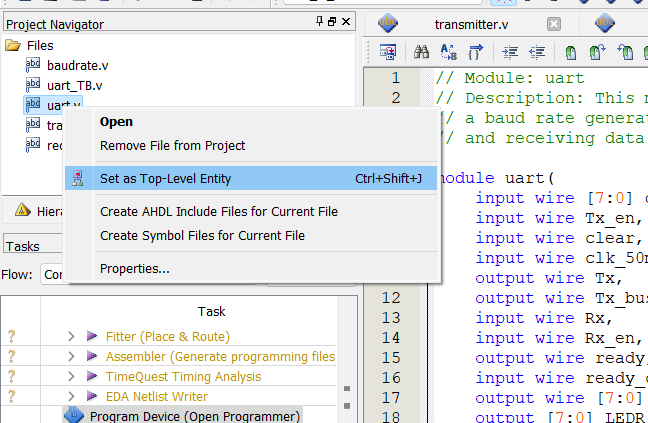
Set the simulation time to 100ns and run the simulation several times.

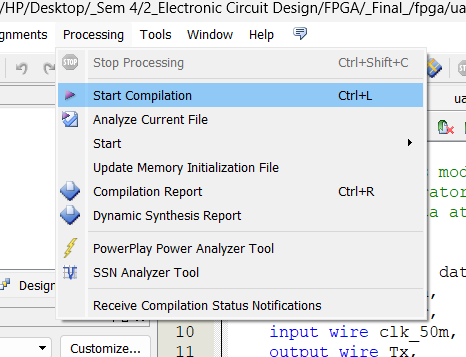


You can experiment with additional test benches and simulate them in a similar manner.

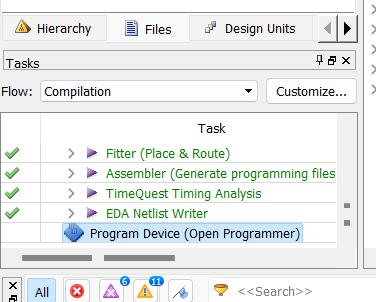
Now it's time to upload the code into the FPGA and simulate its functionality. We've utilized the EP4CE22F17 FPGA from the Cyclone IV E family, so the pin assignments are specific to that. If you're using a different type of FPGA, refer to the corresponding datasheet for pin assignments.

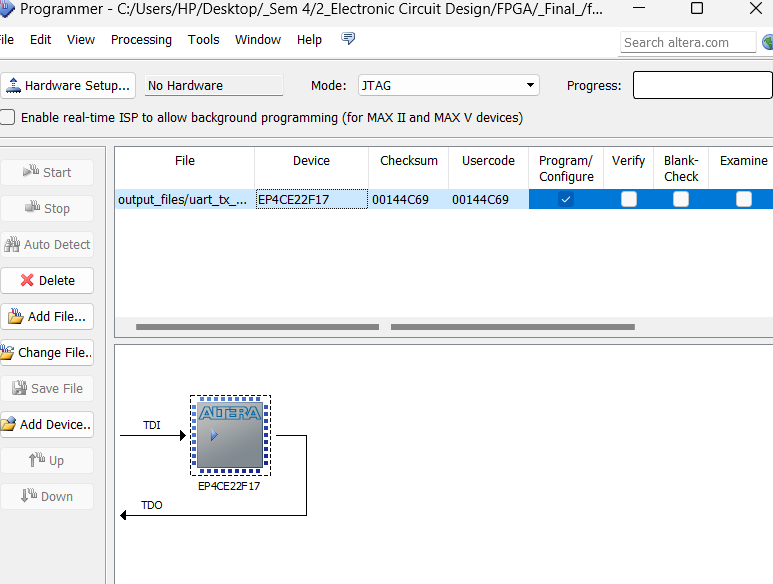
First, set the 'uart' file as the top-level entity and compile it.



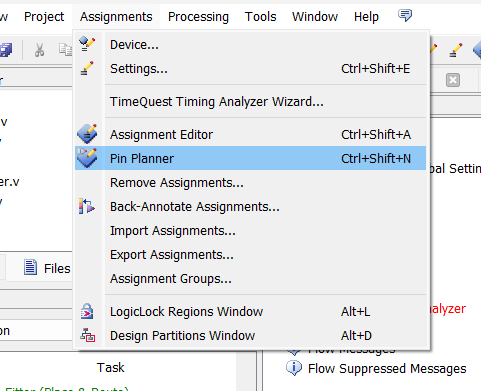


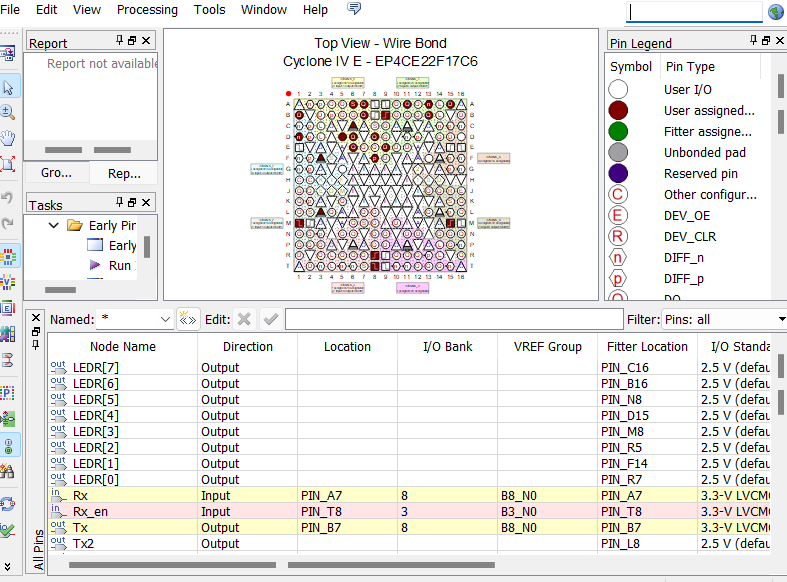
Once the compilation is successful, proceed to the Program Device section and detect your FPGA. Then, upload the code into the FPGA.





For pin planning, follow the steps below. It's essential to assign pins before uploading the code into the FPGA.





Utilize the inbuilt LEDs in the FPGA, as well as the Vcc and Ground pins from the same FPGA for your demonstration.