**Verification Test Plan**

* **Unit Level Testing**
* **Baud rate generator**
  + Random input for the “set\_baud” input which sets the baud rate. Then check against a model. Verifies that it works with a reset.
* **Parity Generator and Checker**
  + Create random input for both the data\_in and the parity select. Measure this against a model to check the operation
* **UART Rx and UART Tx** 
  + For the UART Tx, there is a random parallel input then a model to check the operation against
  + The high level UART testbench tests different inputs while feeding the output of the UART Tx into the input of the UART Rx.
* **Functional and Code Coverage**
  + Using the testbenches explained above, the functional and code coverage was collected using Quartus
* **Testing the UART with the System on Chip**
  + Use a monitor to display the values coming out of the UART\_Tx module and then drive inputs into the UART\_Rx module
  + The assembly code first changes the baud rate, which can be seen on questasim and then ti continues to write “TESTF” in ASCI and then receives it back and sends it to the peripheral again, verifying the operation using the monitor
  + The monitor verifies the UART\_Rx but not the UART\_Tx at the moment due to a faulty model.
* **Formal Verification**
  + Proved the assertions in the system verilog