Norwegian University of Science
and Technology
Department of Electronic Systems

TFE4171 Design of Digital Systems 2 Spring 2019

Semester Project — Part A

Delivery time: Friday 1st March, 23:59

Project description

You, verification engineers for SuperSoC AG, have been tasked with the verification of an untested HDLC (High-level Data Link Control) module to connect two devices together as part of a large project. You are provided with an HDLC implementation and in order to meet a hastily imposed implementation deadline thanks to your well-intentioned colleagues in the marketing department. It is important to retrofit the code with SystemVerilog Assertions to ensure that the design meets the specification before it is integrated.

You are given a lot of flexibility in what assertions you must write, but you must justify each assertion and provide coverage reports generated by the functional and code coverage facilities of SystemVerilog and Questasim. A simple testbench is included which can be used as a starting point.

For Part A a short report of the results is sufficient deliverable.

Grading

The project counts for 20% of the final grade and should be completed in groups of two students, as registered in the beginning of the course. Both students will get the same points. The points will be awarded based on the report delivered and additional analysis of source code and other supplementary materials if needed.

Part A will count for 8%.

Learning outcomes

- Experience working on part of a larger project.
- Adding assertions at module design time.
- Adding assertions to legacy modules.

About:

This exercise will give a short introduction to the term project by writing concurrent and immediate assertions for the Rx module of the HDLC design.

The testbench files are located under the `tb/` directory. `testPr_hdlc.sv` is where you will write your immediate assertions and `assertions_hdlc.sv` is where you will write your concurrent assertions.

1 Immediate assertions

For the immediate assertions you will write assertions to verify correct values in the Rx status/control register and Rx data buffer, both are read from using `AddressIF`. The registers are documented in the HDLC design description.

a) Check for correct values after normal operation in the `VerifyNormalReceive` task.

b) Check for correct values after an abort in the `VerifyAbortReceive` task.

c) Check for correct values after an overflow in the `VerifyOverflowReceive` task.

2 Concurrent assertions

For the concurrent assertions you will write assertions to verify correct behavior of Rx_FlagDetect and Rx_AbortSignal.

a) Write the Rx_flag sequence, which identifies a flag.

b) Write the Rx_AbortSignal property, which verifies that the abort signal is raised when an abort pattern is observed during a valid frame.

Resources

- Questasim User Manual
- Karianne K. Kragseth. HDLC module Design Description. November 15, 2017. (on Blackboard)
- HDLC at Wikipedia.org `https://en.wikipedia.org/wiki/HDLC`
- SystemVerilog Language Reference Manual 3.1a (on Blackboard)
- Modelsim 6.0 Quick guide (contains all the commands used when scripting). `https://users.ece.cmu.edu/~kbiswas/qk_guide.pdf`
- CRC calculation at Wikipedia.org. `https://en.wikipedia.org/wiki/Computation_of_cyclic_redundancy_checks` and `https://en.wikipedia.org/wiki/Cyclic_redundancy_check`