TFE4171 Design of Digital Systems 2

Norwegian University of Science and Technology
Department of Electronics and Telecommunication

Semester Project

Delivery time: Friday 6th May, 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. You must also take advantage of SystemVerilog's constrained randomisation functionality to move away from the directed tests provided in the included testbench. The testbenches also contain some simple assertions. You may use these as a starting point to build more complex assertions using SVA.

The deliverables for the project are 1) a report (not more than 20 pages) which describes the verification approach taken by your team and 2) a zip file containing the assertion code including top-level modules and anything necessary to run the checks.

Specifically, the report should include:

- An introduction to the problem
- A short description of how the HDLC module works
- Your verification methodology:
  - A description of assertions added
  - Assertion and coverage reports with a discussion
  - Corner cases encountered
  - How and when constrained randomisation was used
  - Sources to property checker modules as appendices

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.

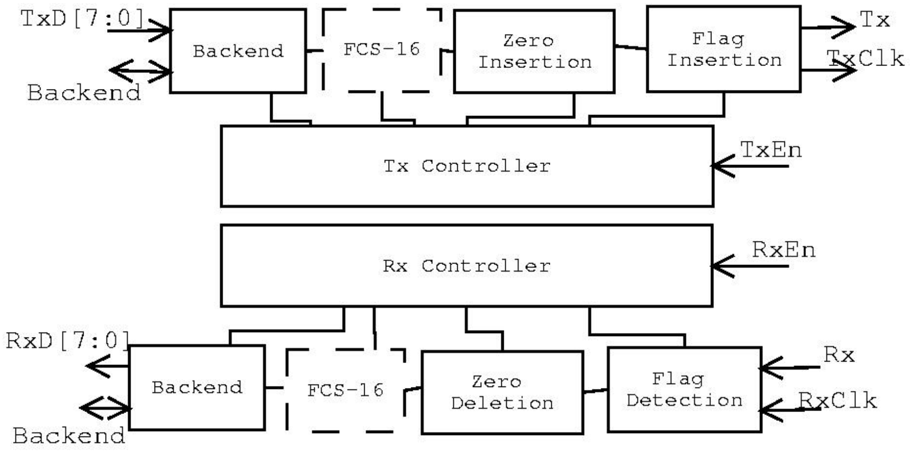

Figure 1: Top-level HDLC diagram.

Learning outcomes

- Experience working on part of a larger project
- Adding assertions at module design time
- Adding/updating assertions to/in legacy modules
- Code (assertions) reuse
- Dealing with complexities associated with the use of multiple HDLs
- Functional coverage
  - Creating a functional coverage matrix
  - Using cover, covergroups, coverpoints, and bins

Getting started

It is important to get started with the project early. The following are general steps you can follow to help begin.

1. Study the HDLC protocol by reading the report in the Resources section below. Supplement your understanding by reading the Wikipedia page.
2. Examine the toplevel testbench in the HDLC project. How is the testbench designed? What ways can you think of to improve it?
3. Familiarise yourself with Questa Sim using the following tutorial: `http://www.eda.ncsu.edu/wiki/Tutorial:Questa_SystemVerilog_Tutorial`
   NOTE: Minor changes are 1) do not copy the tutorial's modelsim.ini 2) use "work" instead of the suggested "mti_lib" and 3) ignore the commands about path setting (e.g., "add questasim63").
4. Follow the instructions below to run the sample testbench.

Running the top-level testbench

This section describes the steps you should take to copy the code and run a simple testbench.

1. Ensure Questa Sim is set up correctly by having completed Exercises 1 and 2 and the tutorial discussed above
2. Copy the term project files from the source directory

   ```
   cp -arv /home/courses/desdigsys2/2016s/dd2master/termproject/hdlc .
   ```

3. Enter the toplevel sources directory

   ```
   cd hdlc/trunk/code
   ```

4. Compile the project by invoking `vsim`

   ```
   vsim -novopt -do top/scripts/model/build_hdlc_top.do
   ```

   Questa Sim should now be open and the project should have been compiled. Check the Transcript for any errors (there should be none; if there are, contact the TA or ask another student for assistance).

5. Select Simulate > Start Simulation...
6. Expand the "work" library
7. Select the testbench "hdlc_tb" from the list
8. Click OK
9. In the Transcript window, type:

   ```
   do top/scripts/model/wave.do
   ```

10. Select Simulate > Run > Run 100
    You should see the Wave pane populated with signals.

Binding a checker to the HDLC module

This section describes how to get started with binding a simple checker to the HDLC module.

1. Enter the toplevel sources directory

   ```
   cd hdlc/trunk/code
   ```

2. Compile the example HDLC properties

   ```
   vlog -sv top/scripts/sva/hdlc_props.v
   ```

3. Create a new file named `top/scripts/sva/sva_wrapper.v`

   ```
   module sva_wrapper;
       ...
   endmodule
   ```

4. Study the bind construct in the resources below, e.g., the SystemVerilog LRM
5. Complete the `sva_wrapper` module by binding the `hdlc_tb` to the property checker above, including signals you wish to write assertions for (e.g., Tx, Rx, TxEN, RxEN, and the associated clocks)
6. Compile the SVA wrapper by invoking `vlog` with the `-sv` flag

   ```
   vlog -sv top/scripts/sva/sva_wrapper.v
   ```

7. Load the simulation

```
vsim -c work.hdlc_tb work.sva_wrapper
```

8. The design should now be loaded. Run the simulation for 100ms:

```
VSIM 1> run 100ms
```

9. You should see something similar to the following output:

```
# ** Warning: There is an ’U’|’X’|’W’|’Z’|’-’ in an arithmetic operand, the result will be ’X
#    Time: 0 ns  Iteration: 0  Instance: /hdlc_tb/DUT/TxBuff
#       8750    RX flag, frame start
#
#     140750    RX flag, frame start
#
#     166250    RX flag, frame start
#
#     166250    TX flag, frame start
#
#     311750    RX flag, frame start
#
#     311750    TX flag, frame start
#
# ** Warning: Data byte 1 missmatch
#    Time: 318610 ns  Iteration: 1  Instance: /hdlc_tb
```

Resources

- Questasim User Manual (on it’s learning)
- Jamil Khatib. HDLC controller core report. April 9, 2001. (on it’s learning)
- HDLC at Wikipedia.org `https://en.wikipedia.org/wiki/HDLC`
- Binding SystemVerilog to VHDL components using Questa (on it’s learning)
- SystemVerilog Language Reference Manual 3.1a (on it’s learning)