Lab 5: Monitors

Goal

Define the monitors for all the DUT's interfaces.

Overview

A monitor is a unit that, as its name suggests, watches an interface and reports any activity on it. For example, on the input interface, the monitor waits for the **packet_valid** signal to become asserted and then catches the bytes being driven on the **data** bus, assembles it into a packet and puts it in a mailbox. The reason why monitors are used instead of just sending the packets from the generator to whichever part of the environment they are needed in is this: suppose the BFM and generator are replaced with an actual design that drives packets to our DUT, if we don't have a monitor on the input interface then we would never know what is being driven, since we relied on our generator to convey this information. One of the main points to consider when creating a verification environment is modularization, along with reusability, so it's a good practice to build its main components to be as independent from each other as possible.

Files

svbt_data_in_monitor.sv
svbt_channel_out_monitor.sv
svbt_include.sv
svbt_environment.sv

Instructions

- Complete the definition of the input **monitor** class in the file **svbt_data_in_monitor.sv**
- Complete the definition of the output **monitor** class in the file **svbt_channel_out_monitor.sv**
- Instantiate and hook up the monitors in the environment class (remember to include them in the **svbt_include.sv** file)
- Compile and run

References

SV_LRM "Assertions"