## Lab 2: The packet class and the first test

### Goal

Learn about **classes**, **randomization** and the **program** module.
Implement the packet class and instantiate it in the test.

### Overview

The DUT accepts input stimulus in the form of packets; a packet is a stream of data that complies with certain rules and has a particular format (described in the functional specification) that is driven to the inputs of the DUT through the interface. The fields that have the keyword **rand** in their declaration will get random values when the function **randomize()** is called on the object, the others will retain their default value.

The **program** module is a clear separator between the test bench and the design, and it provides an entry point to the execution of test benches. The test is a program that is built to create a specific test scenario, by use of constraints, flags, parameters etc. A verification environment can have as many different tests as needed to ensure the full validation of the design.

## Files

*svbt_packet.sv*
*svbt_test_basic_addr_0.sv*

## Instructions

- Complete the **packet** class definition in the file **svbt_packet.sv**, following the comments from the file
- Complete the **test** program from the file **svbt_test_basic_addr_0.sv**; take notice how **new()**, **randomize()** and **post_randomize()** work
- Compile and run

## References

SV_LRM “Classes”
SV_LRM “Aggregate data types”
SV_LRM “Constrained random value generation”
SV_LRM “Programs”