

## Lab 3: Base unit and the environment class

### Goal

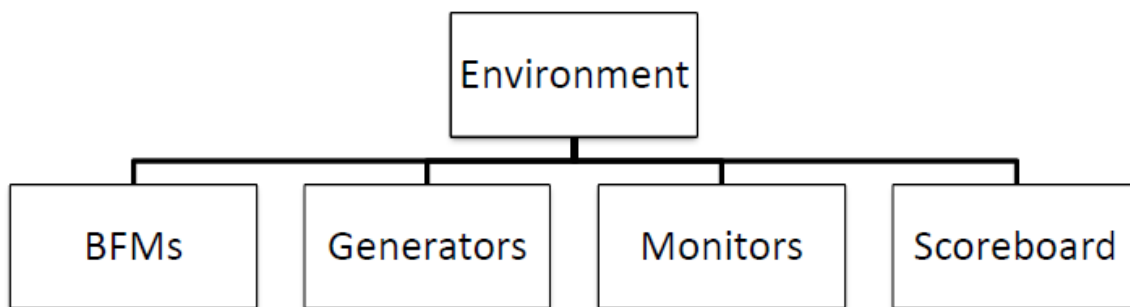
Define the base unit for the environment.

Implement the main environment class that inherits from the base class.

### Overview

It is usual when building complex program architectures to make use of the OOP features of SV, most common of which being the defining of base classes and having the various modules of the environment that share similar characteristics inherit from them.

In this course we will define just one base class and in this lab we will create its first child class, the environment module; this class will be our verification environment's top level module, and all other classes will be instantiated in it. The overall final architecture of the environment will look something like the picture below:



Later on in the course you will see that all the modules in the picture above will extend the same base class as the environment class.

Since the verification environment is starting to be spread out over several files, and their number will keep growing, it is impractical to keep including them in the tests, since every new test will have to have the same (increasing number of) includes over and over. So an **includes** file will be created to centralize all the files, and only this single file will be included in the test.

## Files

`svbt_base_unit.sv`

`svbt_environment.sv`

`svbt_include.sv`

`svbt_test_basic_addr_0.sv`

## Instructions

- Complete the **base\_unit** class definition in the file **svbt\_base\_unit.sv**
- Complete the **environment** class definition in the file **svbt\_environment.sv**
- Populate the **includes** file **svbt\_include.sv** with the paths to all your current files and include it in the test; *in the future remember to add the paths to any new files you write in the include file, otherwise they won't be compiled!*
- Compile and run

## References

SV\_LRM "Assignment statements"

SV\_LRM "Operators and expressions"

SV\_LRM "Tasks and functions"