

Lab 4: Creating a Package

In this lab, you will create the package that works for this module:

```
1 import counter_pkg::*;
2
3 module top;
4
5     counter_if ctr_if();
6     counter DUT(ctr_if.cntr_mp);
7     tester my_tester;
8     checker my_checker;
9     // monitor my_monitor;
10
11     initial begin
12         my_tester=new(ctr_if);
13         my_checker=new(ctr_if);
14         // my_monitor=new(ctr_if);
15         fork
16             my_tester.run();
17             my_checker.run();
18             //my_monitor.run();
19         join_none
20     end
21
22 endmodule // top
```

There are two files already written to make this test bench work:

tester.svh – This contains the definition of the tester class.

checker.svh – This contains the definition of the checker class.

monitor.svh – This file is empty and is the extra credit file.

Your job is to define counter_pkg. The file counter_pkg.sv is already started:

```
1 package counter_pkg;
2
3 endpackage // counter_pkg
4
```

You just have to fill in the middle part to get the package to define the classes.

Use the script run.do to compile and run the simulation.

Extra Credit

When you run this test bench, you'll notice that you can see all the signals in the waveform, but nothing is being printed to the screen. A monitor object could solve this problem.

The monitor object waits for the negative edge of the clock and then prints out the values for `inc`, `ld`, `data_in` and `q`.

If you have the monitor object you'll see output something like this:

```
# Loading work.counter
# data_in: 81 =>   q: 00   inc: 0   ld: 0
# data_in: 63 =>   q: 00   inc: 1   ld: 0
# data_in: 8d =>   q: 01   inc: 1   ld: 0
# data_in: 12 =>   q: 02   inc: 1   ld: 0
# data_in: 0d =>   q: 03   inc: 1   ld: 0
# data_in: 3d =>   q: 04   inc: 0   ld: 1
# data_in: 8c =>   q: 3d   inc: 1   ld: 0
# data_in: c6 =>   q: 3e   inc: 1   ld: 0
# data_in: aa =>   q: 3f   inc: 1   ld: 0
```

For extra credit. Create a class called `monitor` and bring it into the test bench by uncommenting the references to it in `top.sv`. Use the same techniques that were used to create `tester.svh` and `checker.svh`.

You have a head start in file `monitor.svh`:

```
1 class monitor;
2
3 endclass // monitor
```