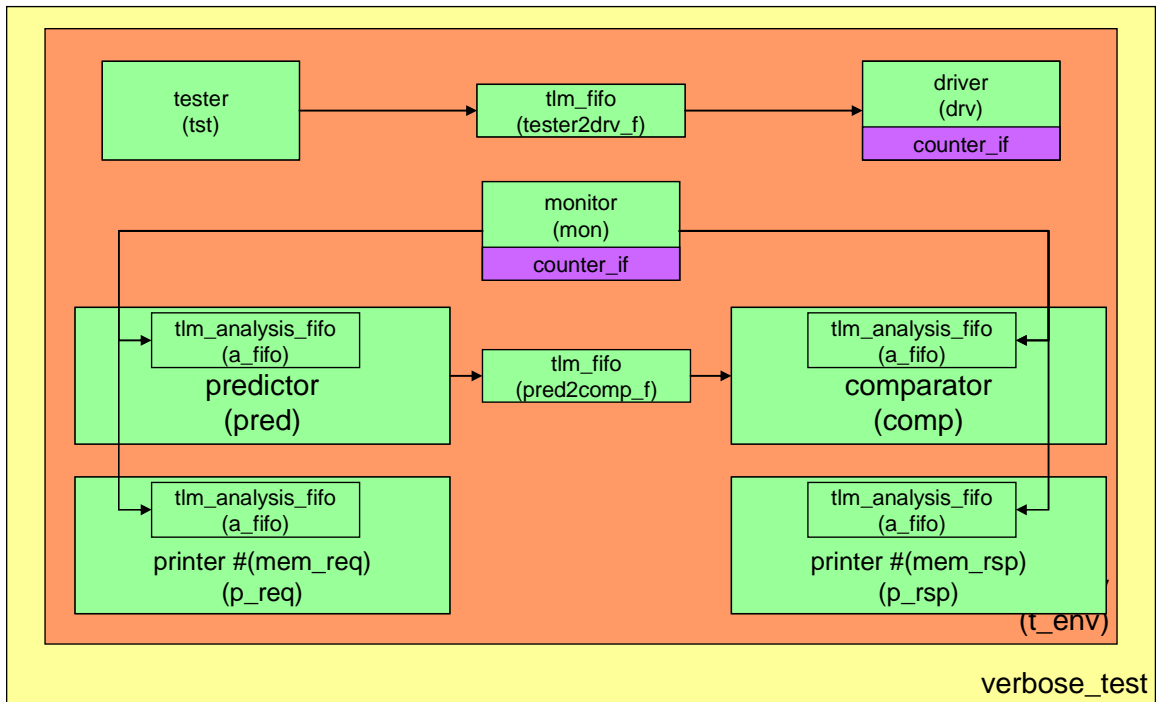


Lab 9: Creating an Analysis Layer

In this lab you will create a transaction level test bench to test the counter that we've used for all the labs since the first one. The TLM test bench in this lab looks just like the one in the lecture, and you can use the examples in the lectures as the basis for building this test bench. The test bench looks like this:



Basic Lab

When you implement the steps in this lab you will be able to run the counter's test bench.

ctr_output.svh

The `ctr_output` object goes from the output of the counter through the test bench.

The monitor creates `ctr_output` objects and puts them onto the `rsp_povm_analysis_port`.

The comparator uses `ctr_output` objects to see if the test bench is working. It calls the `comp` method in `ctr_output` to compare two objects.

1. Complete the `comp()` method in the `ctr_output.svh` file.

comparator.svh

The comparator checks to make sure that the predicted ctr_output object is identical to the actual ctr_output object. In this lab you will do the following to complete the comparator:

1. Declare a `tlm_analysis_fifo` that works with the `ctr_output` object. Call this fifo `actual_f`.
2. `run()` : Call `get()` in the `actual_f` fifo to get an actual result from the test bench. Place the result in a transaction called `actual`.

printer.svh

A printer object will work with any object that has a `convert2string()` function. The object is parameterized so you can tell it what kind of type to use when you declare the object. Please do the following:

1. Declare a class called `printer` and parameterize it to take any type. Call the type parameter `T`.

counter_env.svh

The `counter_env` object instantiates all the other objects and connects them together. Please do the following in `counter_env.svh`:

1. `build()` : Add a call to `create()` that will use the factory to create a new printer object that works with `ctr_output` objects. Call the object `p_rsp`.
2. `connect()` : Connect the `tlm_analysis_fifo` in the predictor to the monitor's request analysis port.
3. `connect()` : Connect the `tlm_analysis_fifo` in the comparator to the monitor's response analysis port.
4. `connect()` : If the `verbose` variable is non-zero connect the request printer's `tlm_analysis_fifo` to the monitor's request analysis port.
5. `connect()` : If the `verbose` variable is non-zero, connect the response printer's `tlm_analysis_fifo` to the monitor's response analysis port.

Running the test

You run the test with the

```
%vsim -c -do "run.do"
```

command:

```
# -----
# OVM-2.0.1
# (C) 2007-2009 Mentor Graphics Corporation
# (C) 2007-2008 Cadence Design Systems, Inc.
# -----
# OVM_INFO @ 0: reporter [RNTST] Running test counter_test...
# OVM_INFO @ 31: ovm_test_top.env.comp [run] passed: data: 00
# OVM_INFO @ 51: ovm_test_top.env.comp [run] passed: data: 01
# OVM_ERROR @ 111: ovm_test_top.env.comp [run] FAILED: Expected: data: 02   Actual: data: 04
# OVM_INFO @ 171: ovm_test_top.env.comp [run] passed: data: 15
# OVM_INFO @ 191: ovm_test_top.env.comp [run] passed: data: 16
# OVM_INFO @ 211: ovm_test_top.env.comp [run] passed: data: 17
```

Running the Quiet Test

To run the quiet test, execute the following commands:

```
% vlib work
% vlog -f compile_sv.f
% vsim -c +OVM_TESTNAME=quiet_test top
VSIM 1> run -all
```

The result is a much quieter test:

```
# -----
# OVM-2.0.1
# (C) 2007-2009 Mentor Graphics Corporation
# (C) 2007-2008 Cadence Design Systems, Inc.
# -----
# OVM_INFO @ 0: reporter [RNTST] Running test quiet_test...
# OVM_ERROR @ 111: ovm_test_top.env.comp [run] FAILED: Expected: data: 02   Actual: data: 04
# OVM_ERROR @ 251: ovm_test_top.env.comp [run] FAILED: Expected: data: 18   Actual: data: 19
# OVM_ERROR @ 311: ovm_test_top.env.comp [run] FAILED: Expected: data: 19   Actual: data: 1c
# OVM_ERROR @ 591: ovm_test_top.env.comp [run] FAILED: Expected: data: 4a   Actual: data: 4c
# OVM_ERROR @ 771: ovm_test_top.env.comp [run] FAILED: Expected: data: d3   Actual: data: d4
# OVM_ERROR @ 791: ovm_test_top.env.comp [run] FAILED: Expected: data: d4   Actual: data: d5
```

Only the errors show.