Questa SystemVerilog Testbench

LAB 2: OOP Basics

Goal Write a more structured Testbench
Get familiar with classes
Randomization
Threads
Mailboxes

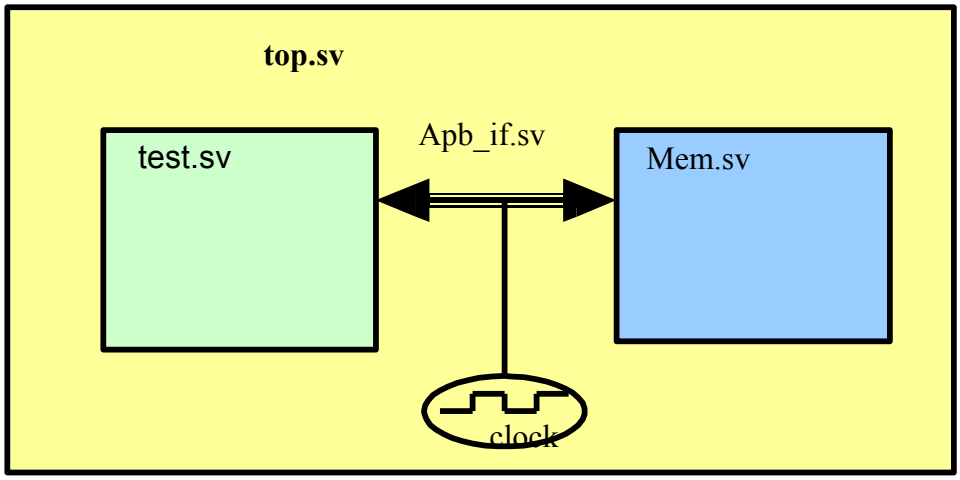
Location From the course Website, download the file lab2.tar.gz
gunzip lab2.tar.gz
tar -xvf lab2.tar

Design APB Interface

Relevant Files in lab2 directory:

`hdl/root.sv`	Global declarations & timescale
`hdl/mem.sv`	APB interface DUT
`hdl/top.sv`	Top level netlist
`tests/test.sv`	Top level testbench program
`apb_env/apb_if.sv`	Interface / clocking block definition
`apb_env/apb_trans.sv`	Transactor class
`apb_env/apb_gen.sv`	Generator class
`apb_env/apb_master.sv`	Bus master class

Testbench Environment



Steps to hook up a DUT to a Testbench

1. Create apb_if interface with modports and clocking blocks
2. Create testbench program
3. Create top module
4. Compile and run

Lab instructions:

1) Complete the transactor class **`apb_trans`** in **`apb_env/apb_trans.sv`**. Create 3 random variables, create method "**`display`**" to print out the random data, and create a method **`copy`** that returns a copy of the class data.
2) In **`apb_env/apb_gen.sv`**, complete the generator class **`apb_gen`**. Write the constructor for this class that initializes the mailbox, initializes the transactor object, and initializes the max number of transactions. Create the task **`main`** with a loop that sends **`max_trans_cnt`** random transactions to mailbox.
3) In **`apb_env/apb_master.sv`**, edit the constructor function **`new`**, and initialize the mailbox and create an APB transaction **`tr`**. In the **`main`** task, create an infinite loop to get transactions from the mailbox, decode them, and execute them with the **`read`**, **`write`**, and **`idle`** tasks.
4) Complete the main program in **`tests/test.v`**. Initialize the mailbox, call the constructor for the generator and the master, call the **`main`** & **`reset`** method of the master object, and call the **`main`** method of the generator object. Lastly, end the test when the generator ends, the mailbox empties, and then a few cycles more.

** Search for "**`LAB`**" in the lab files to see where to add your code.