System Verilog Lab1

After completing this lab, you should be able to:

Understand the design specification.

Implement function/tasks

Generate and Drive stimulus into DUT

Verify DUT behaviour with waveform.

LAB1:

Step 1: Declare the necessary variables and nets for the DUT (ram_dut) in testbench.

1. Open testbench.sv
2. Define required signals (clk,reset,wdata,…..). Add this code in Section 1 in the testench.sv

Step 2: Instantiate DUT (ram_dut) in testbench along with ports connection.

1. Instantiate ram_dut with instance name dut_inst along with port connection. **Use name based port mapping**.
2. Add the below code in Section 2 in testbench.sv

   ram_dut dut_inst(.clk(clk),.rst(rst),.....);

Step 3: Generate clock and reset in testbench.

1. Initialize clk to 0 in initial block and generate clock.
2. Add the below code in Section 3 in testbench.sv

```
initial clk=0;
always #5 clk = ~clk;
```

3. Write a task named reset_stimulus to apply reset to DUT.
4. Add the below code in Section 4 in testbench.sv

```
task reset_stimulus();
        $display("[Testbench] Reset applied to DUT");
        rst <= 1;
        @(posedge clk);
        rst <= 0;
        $display("[Testbench] Reset completed");
endtask
```

Step 4: Write a task write stimulus for writing into the memory.

1. Define task with address and data as arguments with input direction.
2. Add the below code in Section 4 in testbench.sv

```
task write_stimulus(input reg[2:0] addr_t,reg [7:0] wdata_t);
        wr<=0;
        wdata<=wdata_t;;
        addr<=addr_t;
        @(posedge clk);
endtask
```

Step 5: Write initial block to start the verification flow.

1. Implement initial block and call all methods you have implemented so far.
2. Add the below code into Section 5 in testbench.sv

```
initial begin
    reset_stimulus();

// Write task (address,wdata)
    write_stimulus(0,10);
    write_stimulus(1,20);
    write_stimulus(3,30);
    write_stimulus(4,40);

// Wait for the DUT to process the last transaction
    @(posedge clk);

    $finish;
end
```

Step 6: Validate the output of DUT with waveform.

1. Use +acc while compiling and do wave.do in the do file.

```
vlib work
vlog ram_dut.sv testbench.sv +acc
vsim work.testbench
do wave.do
run -all
```

Step 7: Simulating the design

1. Navigate to the Lab 1 directory copy the path and open the simulator.
2. Use cd {../Lab1}
3. do run.do
4. Verify the waveform below.