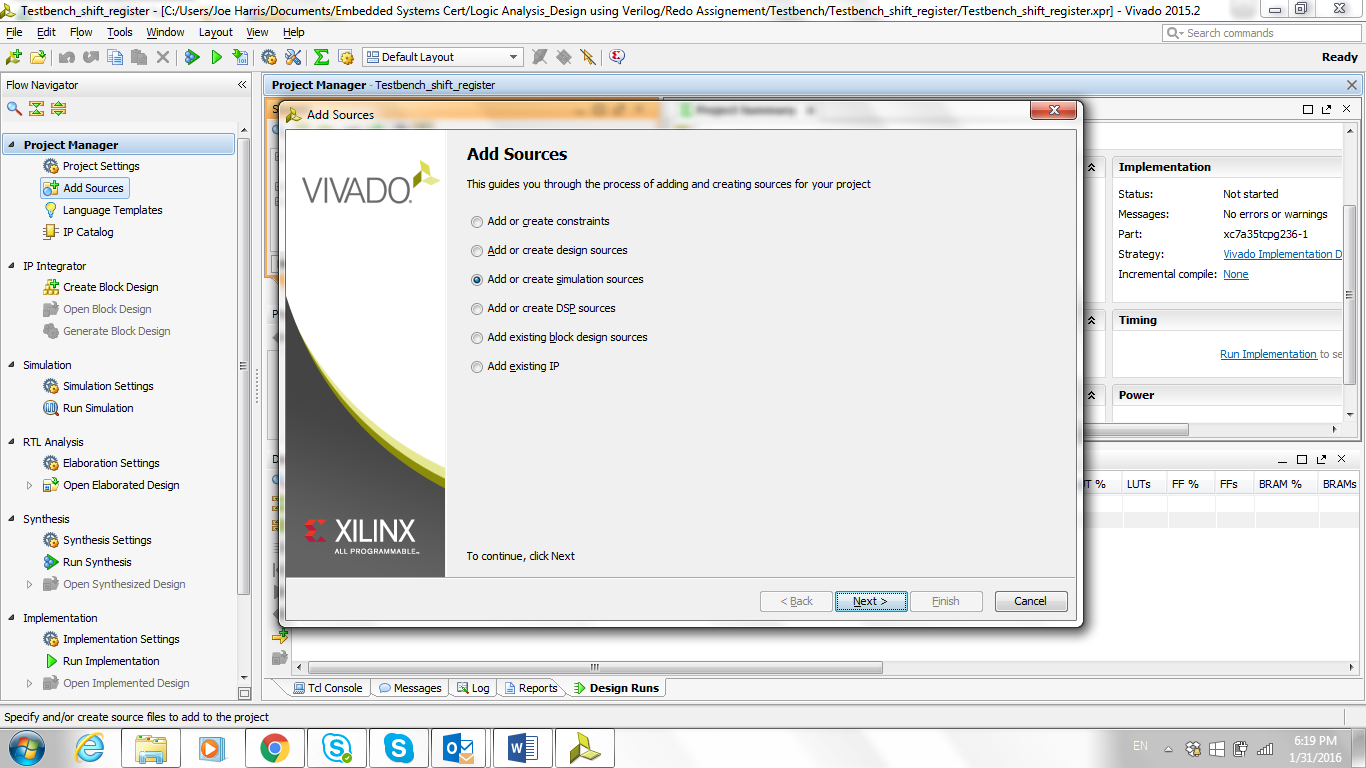
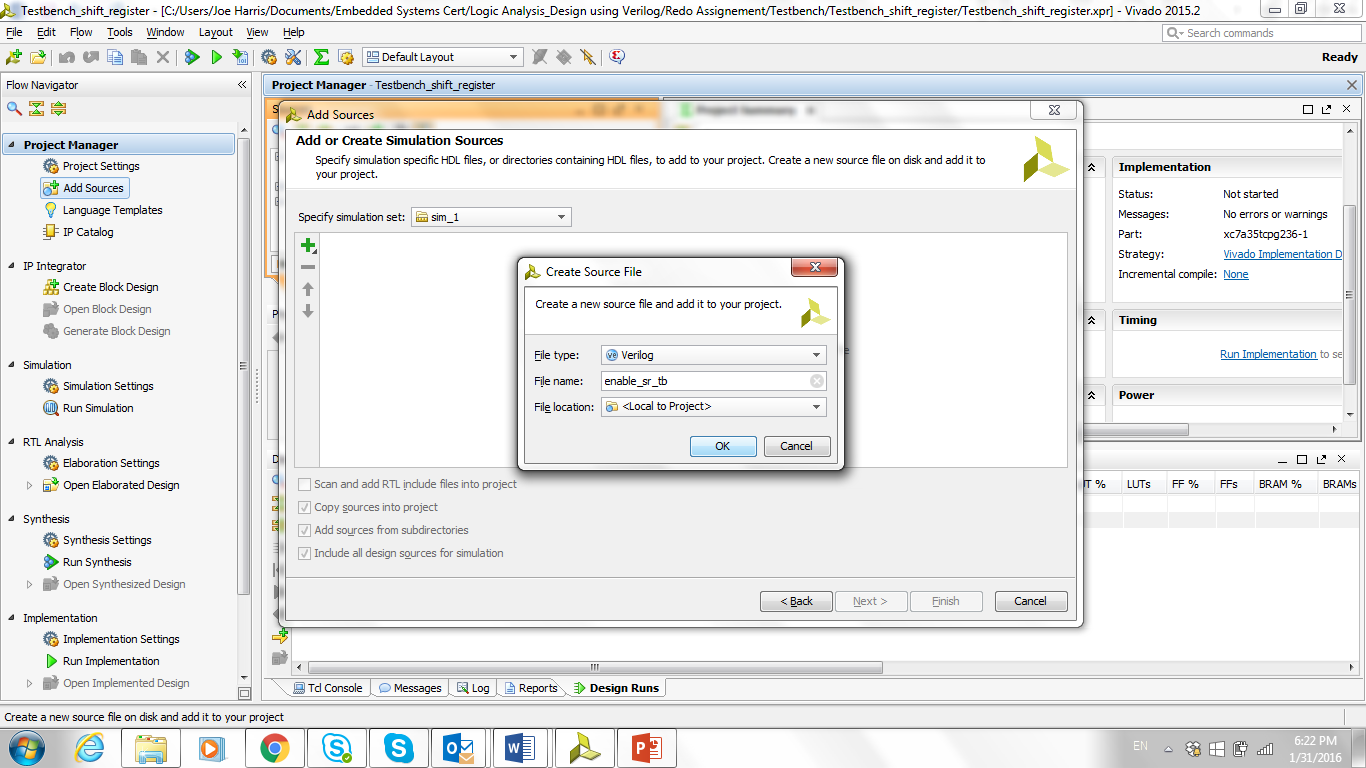
Step 1: Add Sources and choose “Add or Create Simulation Sources



Step 2: Create file called enable\_sr\_tb



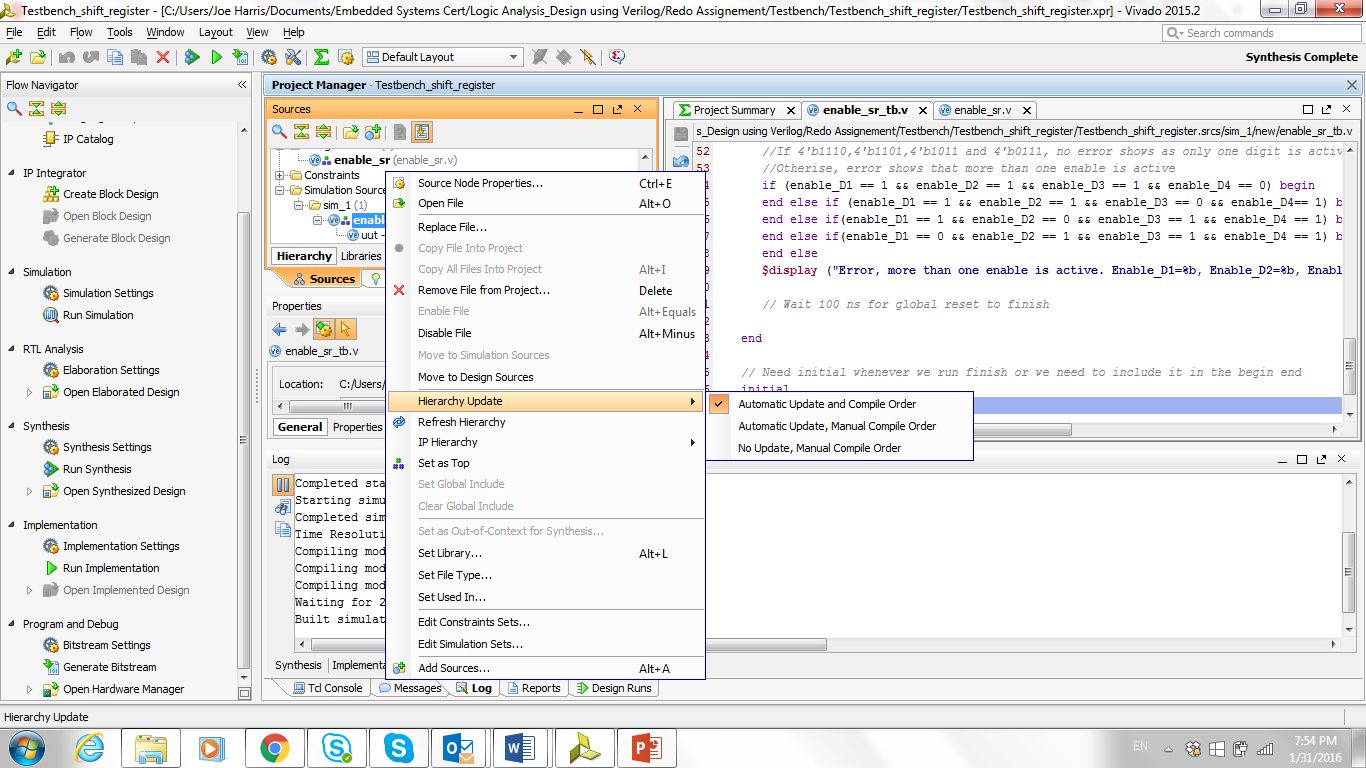
Step 3: Create testbench file.

1. Import the module enable\_sr from stop watch. That is the file we want to simulate
2. Create testbench module enable\_sr\_tb();
3. Key in inputs and outputs of the module enable\_sr(). Remember the inputs for enable\_sr is now in register type while the outputs become net type.
4. Instantiate the unit under test (uut) which is the enable\_sr
5. Generate clock which period (T) is 20ns
6. Use the conditional statement to create error checking system. In this example, we want to check whether there are more than one digits are active.

Note: In the original enable\_sr() file, we should initial the pattern as 4’b0011 so that there are two digits are active to create error.

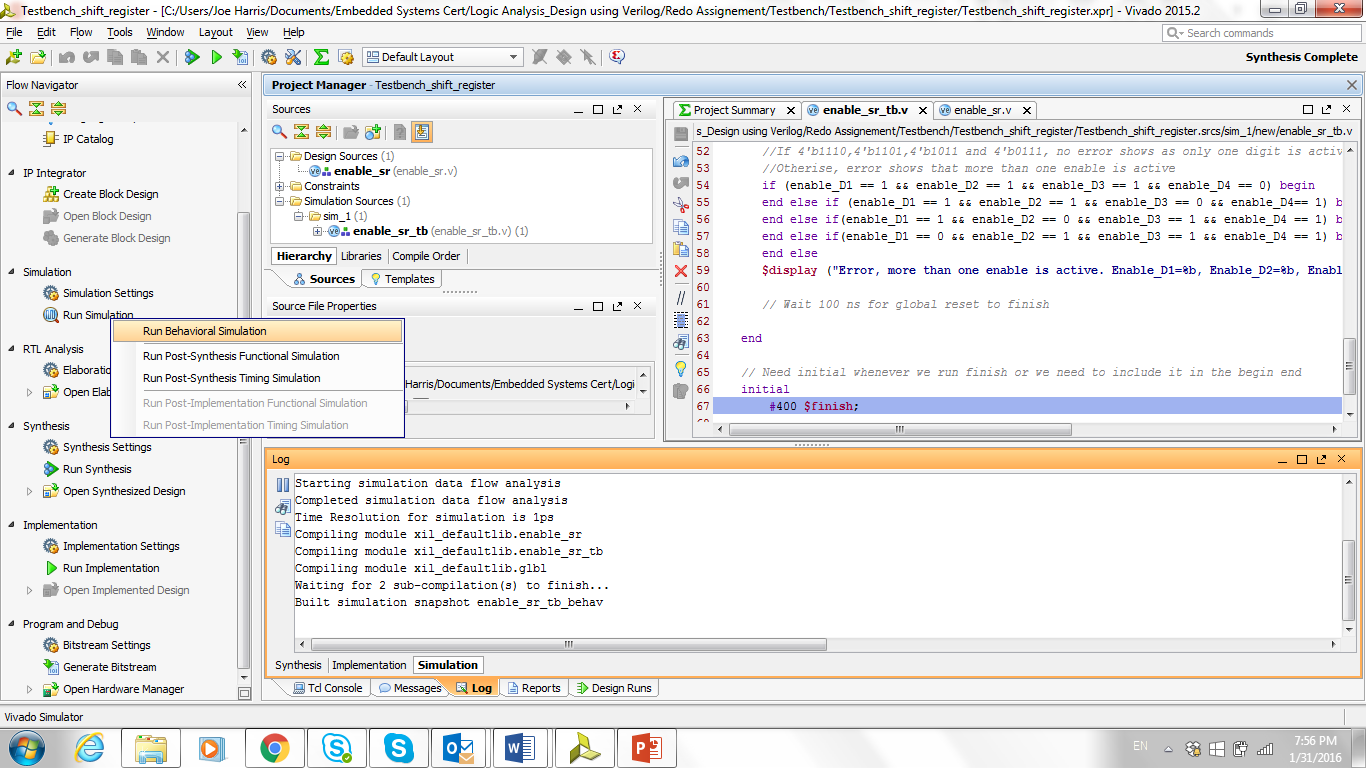
1. Use system task $display to show the error
2. Use system task $finish to complete the simulation at time 400ns

Step 4: Set the enable\_sr\_tb as the top level under the simulation



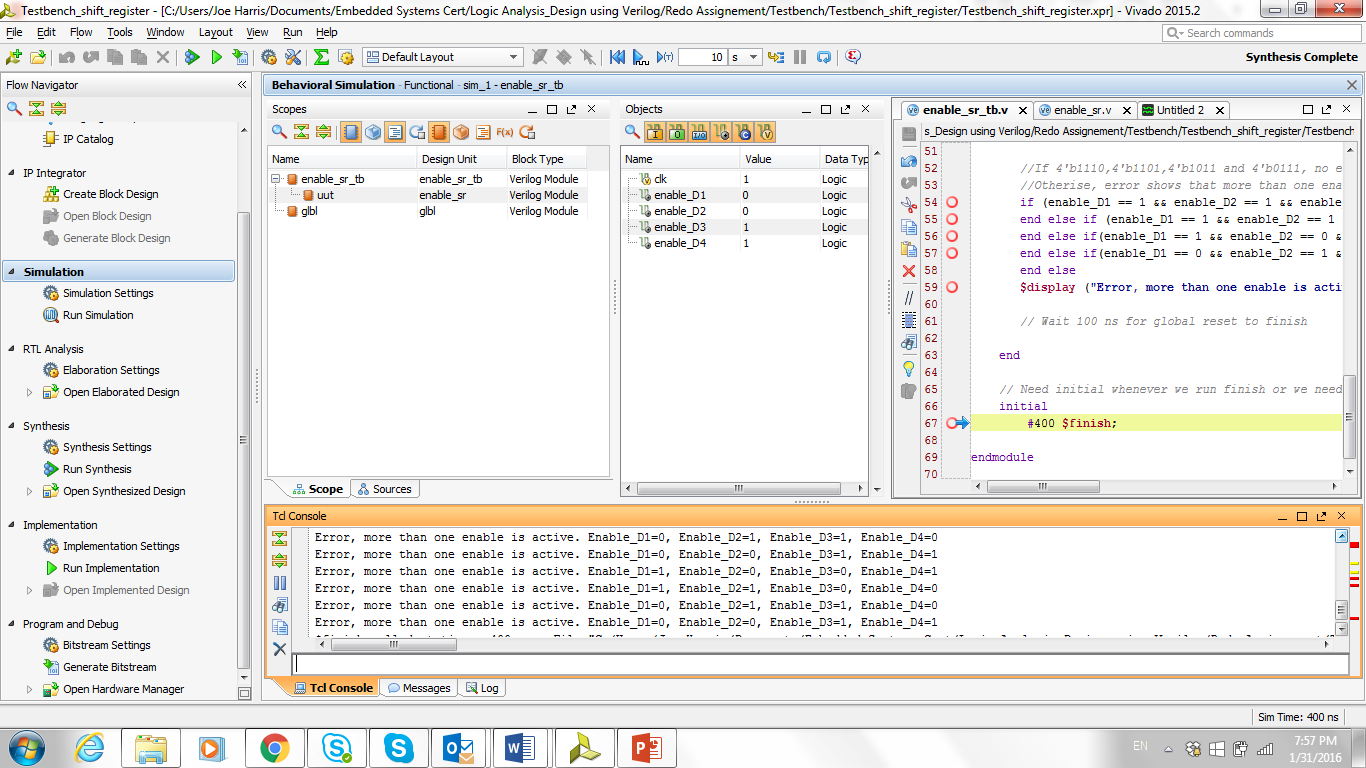
Step 5: Run synthesis

Step 6: Run behavior simulation

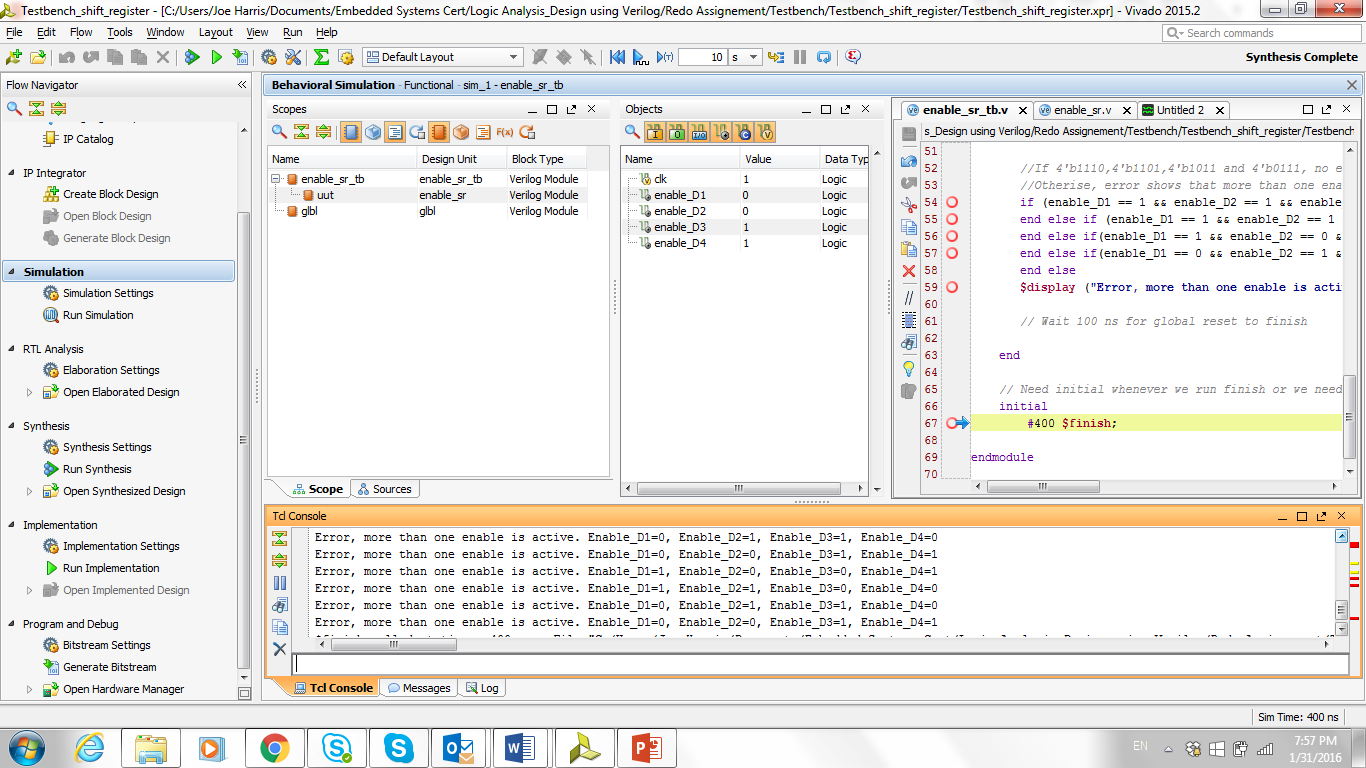


Step 7: Generate Simulation Result

You will see the simulation windows. It contains different panels.



You will see the error message in the console panel



You can also see the waveform in the scope

