**Introduction to SystemVerilog HDVL**

**Lab Manual**

**Lab 0 and Lab 1: SystemVerilog Immediate Assertions**

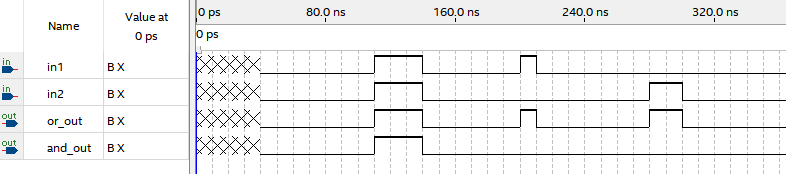
**Objective: *The objective of this lab is to get introduced to***

* **SystemVerilog Immediate Assertions**
* The SystemVerilog code and the corresponding testbench for the 4-bit ALU is present in the SV\_assertions folder [lab0].
* Assertions are inserted into the RTL code for checking the valid inputs and the valid select lines.
* Assertion will be reported as failure when the inputs or the select lines are unknown.

Lab0:

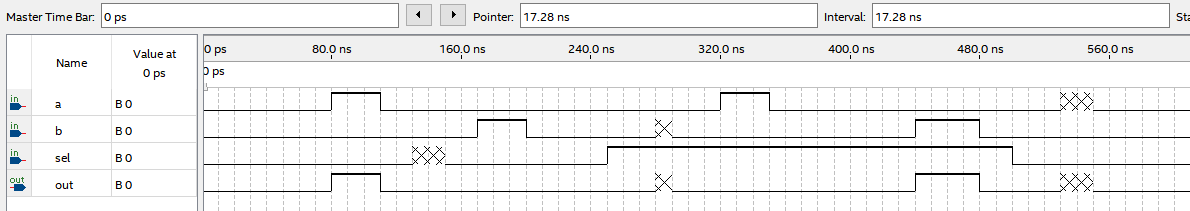
Step 1:

Inspect and\_or\_assert.sv in the lab0 folder. Based on the waveform below, is the assertion met? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



Step 2:

Inspect mux\_2\_1\_assert.sv. Based on the waveform below, is the assertion met?



Is the assertion met? When does it fail?

Explain below.

Step3:

Inspect mux\_4\_1\_assert.sv. Add the assertion that checks that makes sure at not time during the simulation does any input go to the “X” state.

Paste your modified mux\_4\_1\_assert.sv below.

**Open lab1 folder.**

Inspect assertion\_ALU.sv.

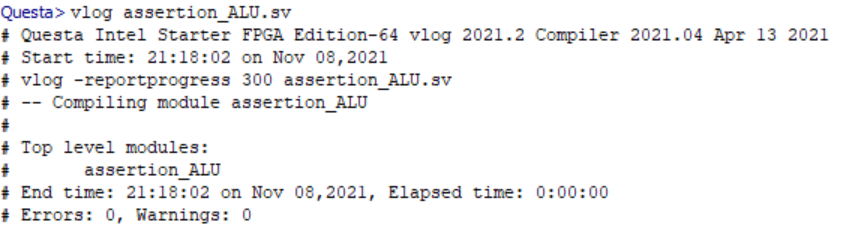
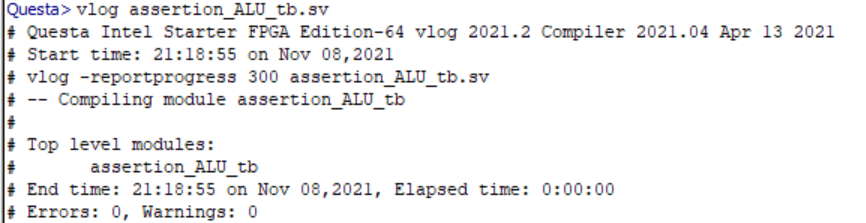
Explain if you can detect when there is either an “X” on signals x or y. Is it possible to detect from the assertions which signal fails the condition?

Inspect assertion\_ALU\_tb.sv.

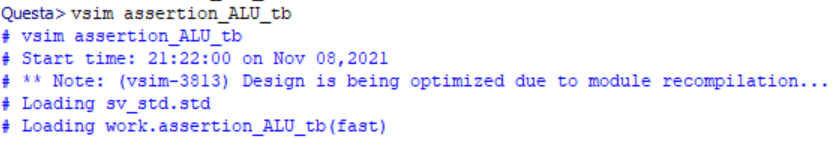
What time(s) would you expect the assertions declared in assertion\_ALU.sv to fail and display in the log?

**Step 1:** Compile the given assertion\_ALU.sv and assertion\_ALU\_tb.sv files using Questasim. (To launch Questasim, type Questasim in the search. The appendix indicates download instructions).

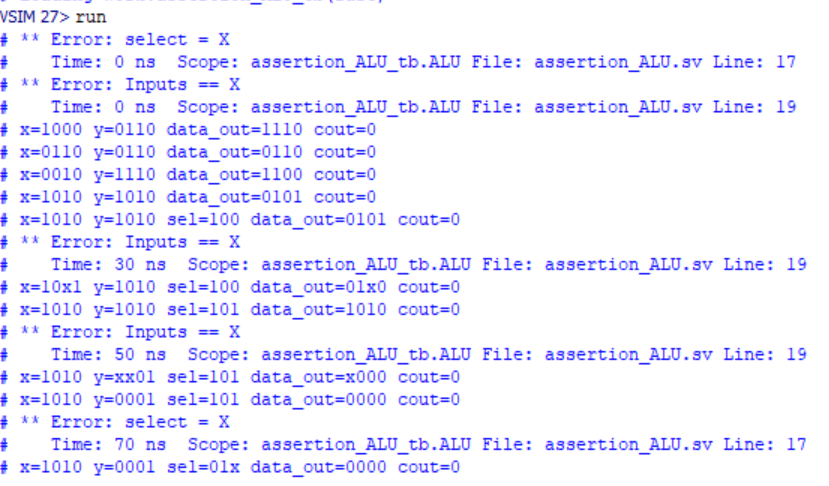
**\***Should you not have Questasim installed, an excellent alternative to Questasim is edaplayground.com. This site offers a number of EDA tools that you can access for free. Enter the design code on the right panel and the testbench on the left panel. Note that the EDA Playground defaults to compiling testbench.sv and design.sv, so if you load the files with the “+” icon, you need to copy them to the testbench.sv and design.sv tabs. Once loaded, hit the Run button on the upper right and observe the log output on the bottom screen.



**Step 2:** Simulate the testbench



**Step 3:** run



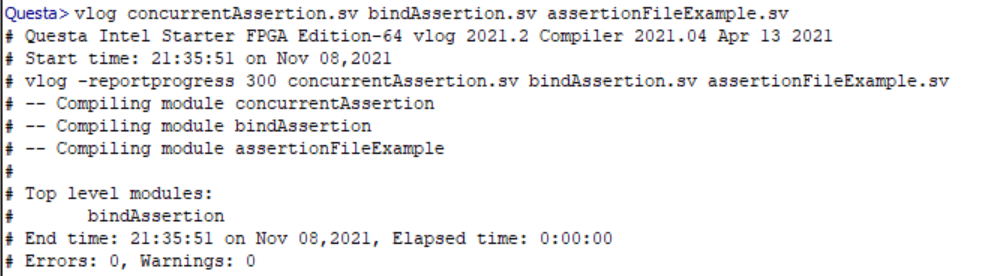
**Lab 2: Concurrent Assertions**

**Objective: *The objective of this lab is to get introduced to***

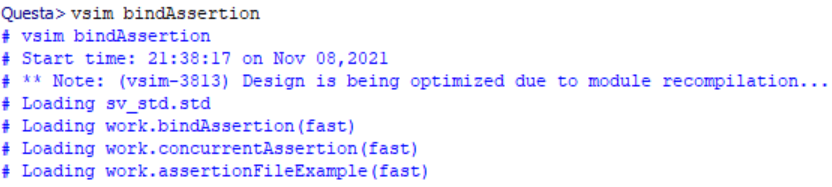
* **SystemVerilog Concurrent Assertions with the bind operator**
* The SystemVerilog code, assertion properties file and the testbench is provided for a 4-bit counter [lab2].
* Two assertion properties are verified for validating the functionality of the counter.
* **Perform Simulation using Questa- Intel FPGA Starter edition.**

**Step 1: Compile the given SystemVerilog files after analyzing the properties**

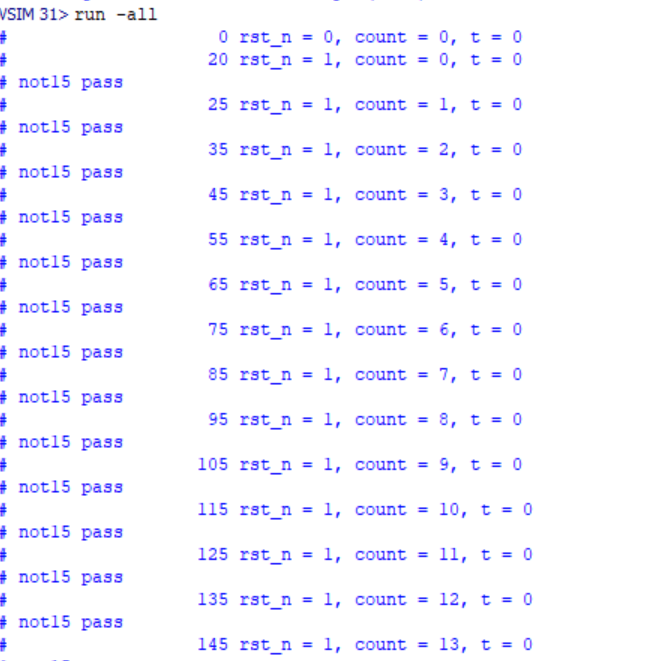
In the lab 2 folder, files**.** Inspect all the three SystemVerilog files

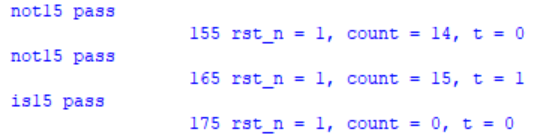
* concurrentAssertion.sv
* assertionFileExample.sv
* bindAssertion.sv (note this one is your testbench)

Compile the files using vlog

 **Step 2: Simulate the testbench**

**Step 3 : Run**





Bonus: Create a design with assertions and associated testbench that demonstrates the $rose and $fell System Verilog constructs.

**END OF LAB EXERCISE**

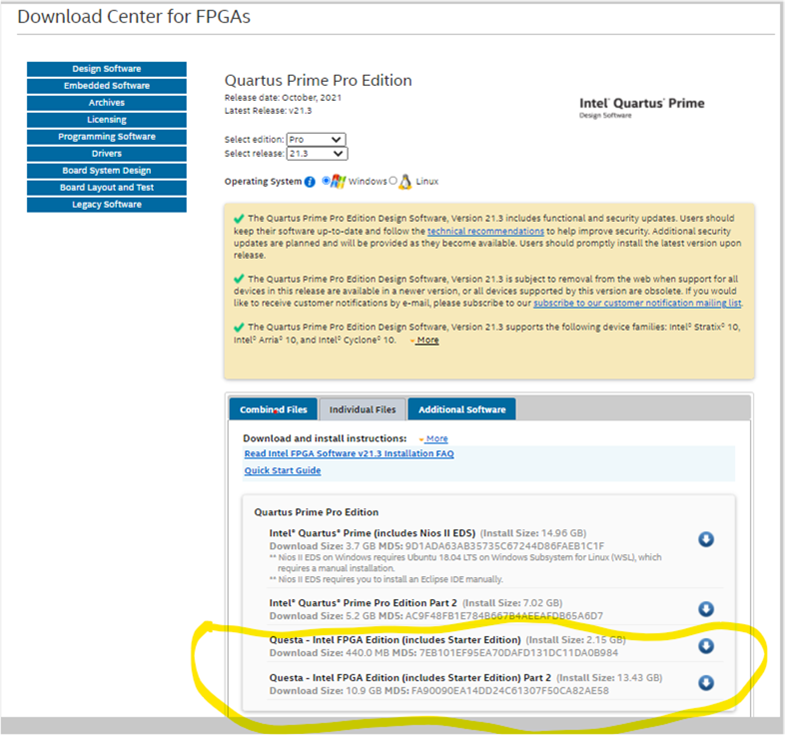
**Appendix**

**Questa - Intel FPGA starter edition – Installation details**

If you don’t already have the Questasim simulator, you will need it to run the labs.

<https://fpgasoftware.intel.com/21.3/?edition=pro&platform=windows>

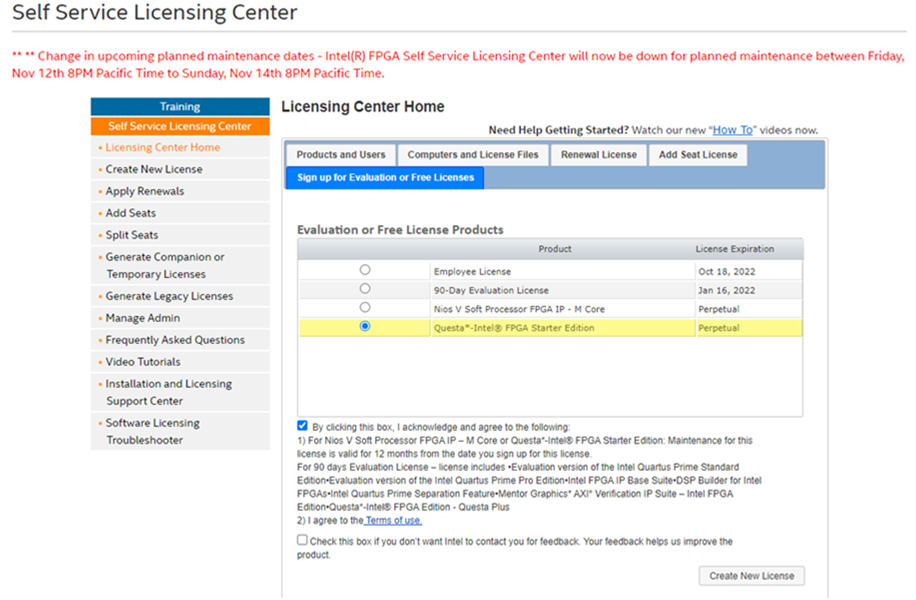
Download the files highlighted:



To obtain a license:

[https://fpgasupport.intel.com/Licensing/license/index.html#](https://fpgasupport.intel.com/Licensing/license/index.html)

Sign-in as an employee.



Click on create new license, follow the instructions on machine name and MAC address. You will receive an email with a new license from [authorization@intel.com](mailto:authorization@intel.com) .

You will need to edit/add environment variables Search 🡪 environment variables. Add as shown below.

