**Introduction to SystemVerilog HDVL**

**Lab Manual**

**Lab 1: SystemVerilog data types and associations**

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

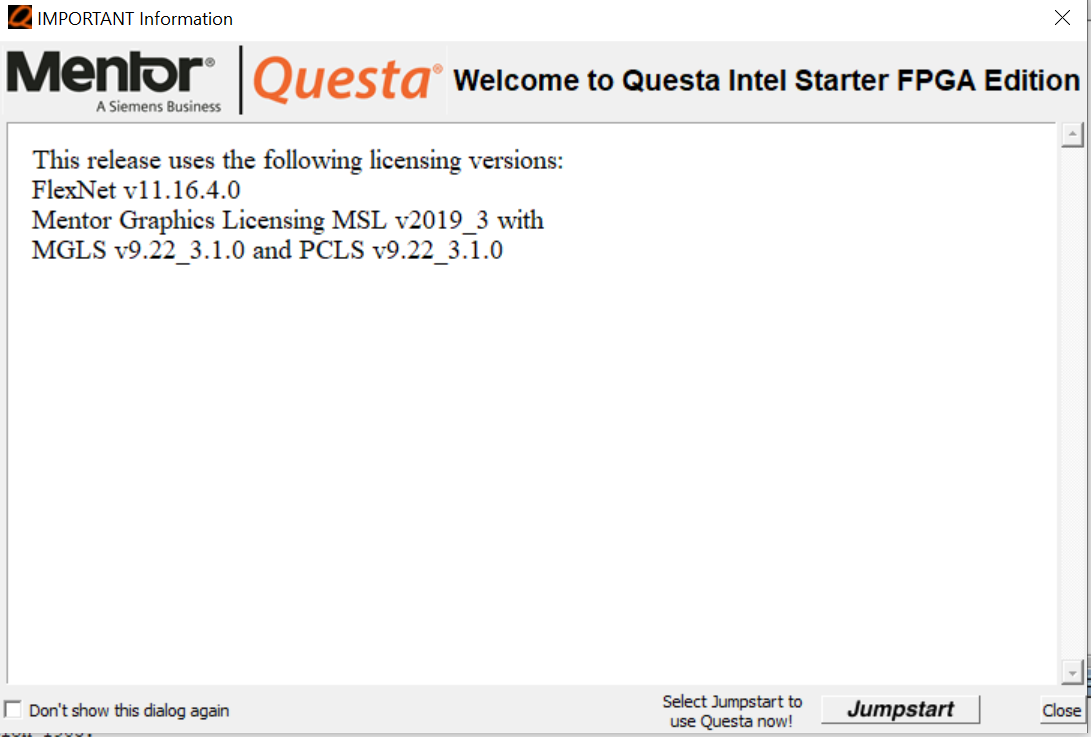
* **SystemVerilog logic data type for both design and testbench.**
* **always\_comb specialized always procedural block**
* **SystemVerilog Implicit .name and Implicit .\* associations**
* **SystemVerilog constraint random test case generation**
* **Check the functionality of the design using Questa- Intel FPGA Starter edition.**

**N-bit ALU which performs 8 operations:**

The SystemVerilog source code for the 4-bit ALU which performs both arithmetic and bitwise operations along with the testbench file has been already created for you to check the functionality using QuestaSim.

**RTL Simulation of the ALU design using Questa- Intel FPGA Starter edition**

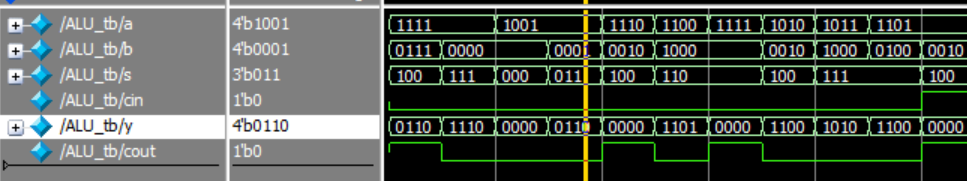
1. Launch the **Questa-Intel FPGA starter edition** from the Windows start menu
2. Close the Questa introductory window

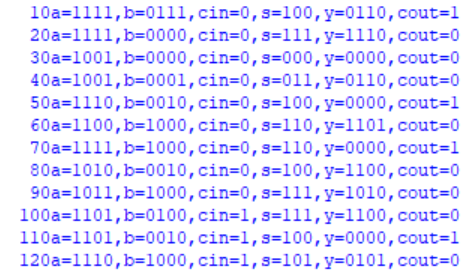


1. Set the project directory. From the QuestaSim File menu, select Change Directory. Browse to the location <Project Directory>\lab1
2. *A Questasim macro file called a .DO file has been created for you. This file named* ***alu\_tb.do*** *contains all of the steps to run the Questa tool and perform simulation.  
   This includes:*a. *Creating a working library using the* ***vlib*** *command*b. *Compiling all of the SystemVerilog files into the working library with the* ***vlog -sv*** *command.*c. *Loading the simulator with the top-level testbench file using the* ***vsim****command.*d. *Opening the waveform window with the* ***wave*** *command.*e. *Adding target signals to the wave window (and formatting them) using the* ***add wave*** *command.*f. *Advancing simulation using the* ***run*** *command.  
   You may open the* ***alu\_tb.do*** *file in a text editor if you wish to view the specific  
   commands used.*
3. Execute the macro file. In the console window execute the do file by typing

**do alu\_tb.do**

1. Check the simulation results for correct functionality





* End your simulation. From the **QuestaSim Simulate** menu, select **End Simulation** OR type **quit –sim** in the **Questa Transcript** window.
* You will notice that when you select commands from the Questa pull down menus, the command is displayed in the transcript window.
* You can also use the up/down arrows to select prior commands to execute.

**END OF EXERCISE 1**

**Lab 2: FSM Modeling using Enumerated Data types**

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

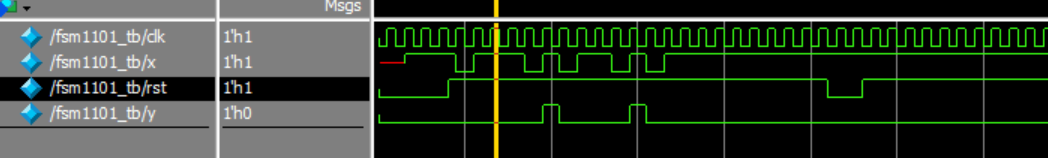
* **SystemVerilog Enumerated data type for modeling FSMs.**
* **always\_ff specialized always procedural block**
* **Named begin and end block**
* **Free running clock generation using SV increment operators**
* **Check the functionality of the design using Questa- Intel FPGA Starter edition**

**RTL Simulation of the Mealy 1101 Pattern detector**

1. Set the project directory. From the QuestaSim File menu, select Change Directory. Browse to the location <Project Directory>\lab2
2. *A Questasim macro file called a .DO file has been created for you. This file named* ***fsm1101\_tb.do*** *contains all the steps to run the Questa tool and perform simulation.*
3. Execute the macro file. In the console window execute the do file by typing

**do *fsm1101\_tb.do***

1. Check the simulation results for correct functionality



* End your simulation. From the **QuestaSim Simulate** menu, select **End Simulation** OR type **quit –sim** in the **Questa Transcript** window.
* Convert the given Verilog source code into SystemVerilog and check for functionality with the following enhancements
* Copy the fsm1101.v and fsm1101\_tb. v Verilog files and save them with .sv extension

Modify the Verilog codes into SystemVerilog with the following enhancements

* + - Convert all the reg and wire data types to logic
    - Use enum in place of parameters
    - Replace the Verilog always blocks with the SystemVerilog specialized always blocks [ always\_comb and always\_ff]
    - Named begin-end block
    - Apply Implicit .\* associations in the testbench instantiation
    - Generate free running clock with SV increment operator
* The SystemVerilog solution is provided in the **lab2\_solution** folder for reference. It is recommended to try it on your own to get a better experience.

**END OF LAB EXERCISE 2**