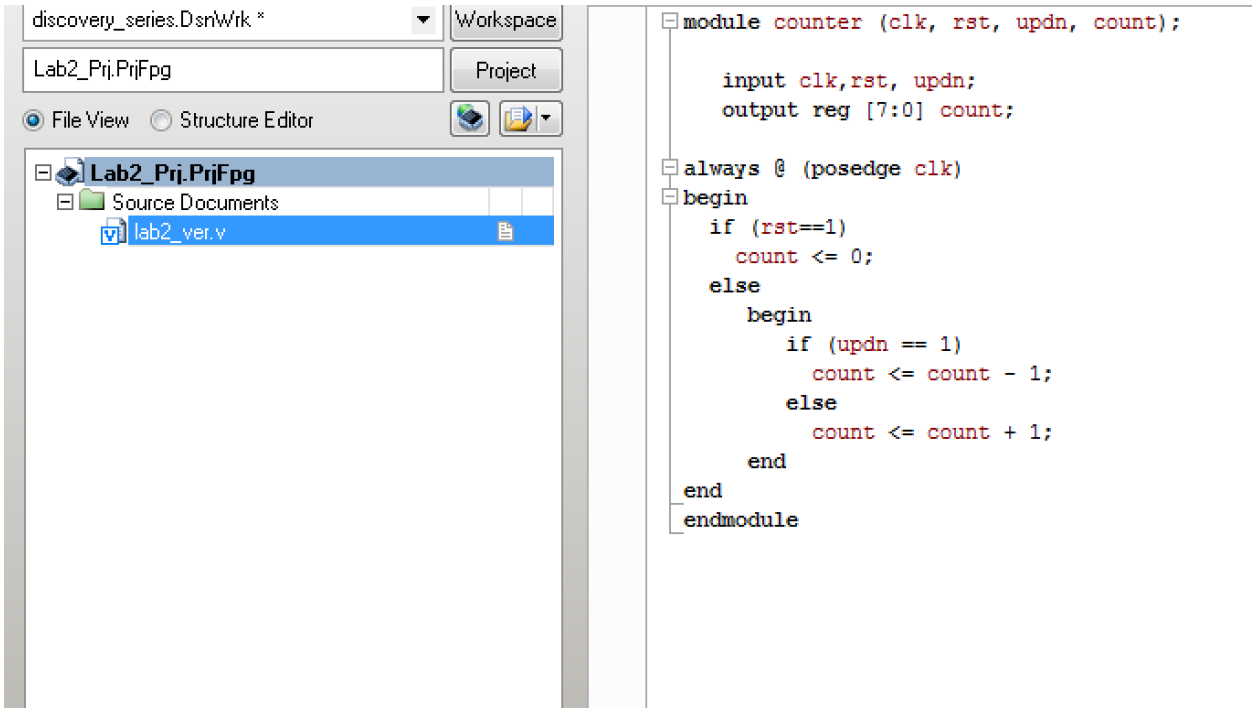
Lab2 – Up Down Counter using HDL Entry

RMB → Right Mouse Btn **MMB → Middle Mouse Btn** **LMB → Left Mouse Btn**

RHS → Right Hand Side **LHS → Left Hand Side**

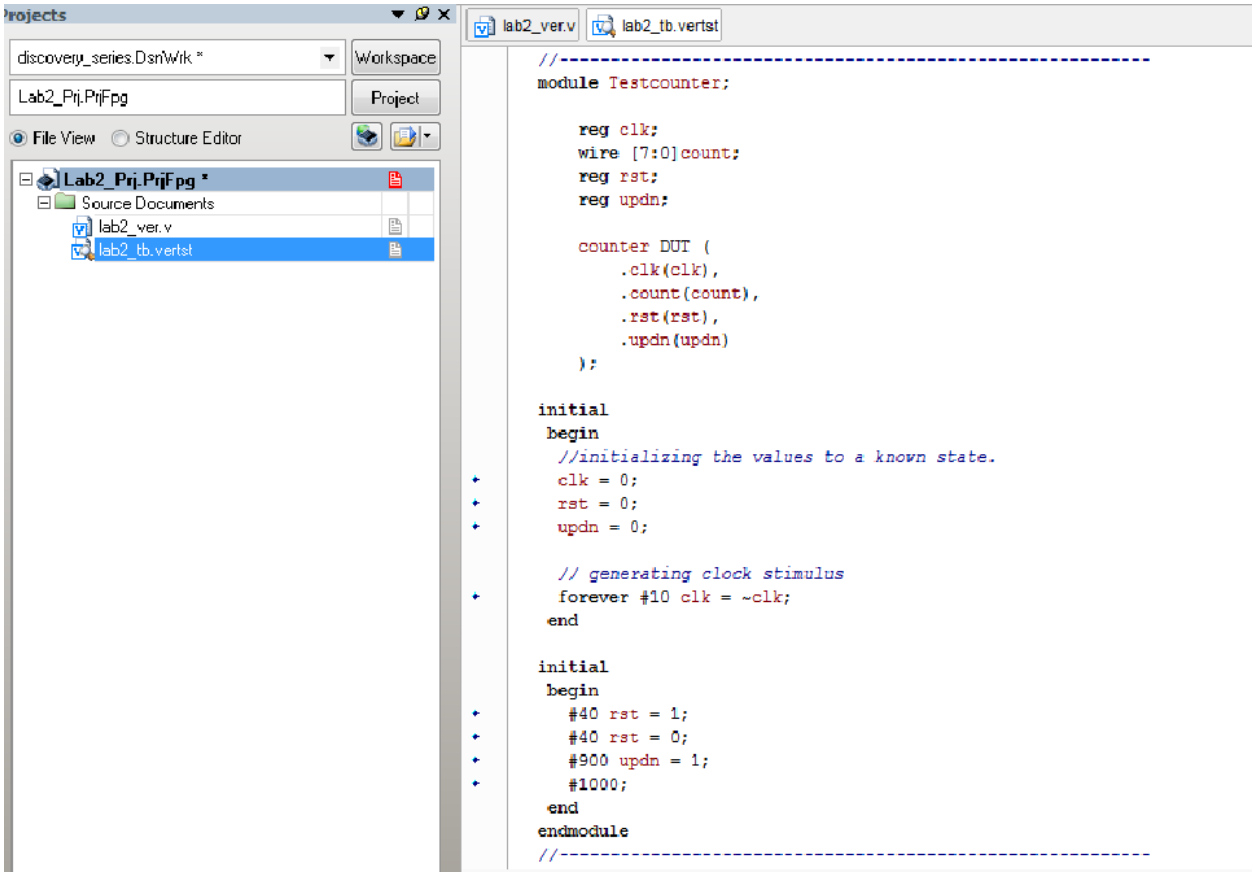
1. Open Altium Designer R10 clicking on Start → Programs → Altium → Altium Designer Release 10
2. From the menu bar select File → New → Project → FPGA Project.
3. From the Projects window, select the newly created Project **FPGA_project1.PrjFPG,** RMB → Add New to Project → Verilog Document.
4. RMB on **FPGA_Project1.PrjFPG → Save Project As** and choose the location to **D:\training\lab2**, save the verilog document as **lab2_ver** and the project as **Lab2_Prj**
5. Now type the below code in the lab2_ver.v file as shown below.



6. After successful HDL entry, compile the project to check for syntax errors. Once the project is compiled successfully, proceed for Functional Simulation by following the below steps.

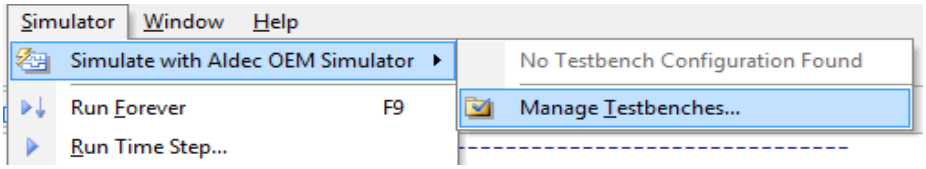
Functional Simulation

7. From the Menu Bar select **Simulator → Create Verilog Testbench.** Altium Designer will automatically generate a verilog testbench template for the design counter. Now add the stimulus according to the design requirement and save the testbench document as **lab2_tb.VERTST** as shown below.

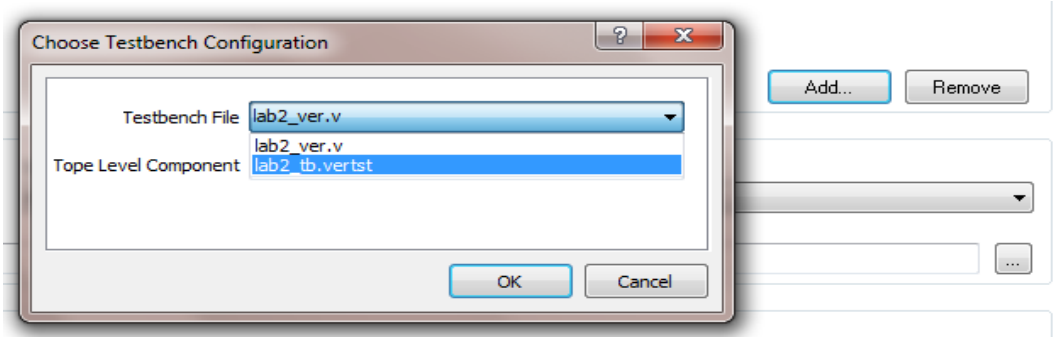


8. Save the Project by **RMB** on Project **Lab2_Prj.PrjFPG → Save Project.** Compile the Project by **RMB** on Project **Lab2_Prj.PrjFpg → Compile FPGA Project Lab2_Prj.PrjFpg**

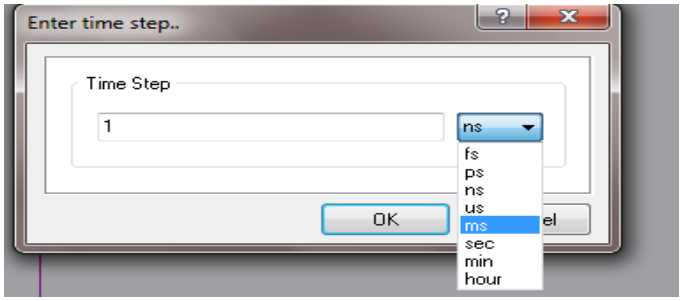
9. After Successful Compilation of the Project, select the Menu **Simulator → Simulate with Aldec OEM Simulator → Manage Testbenches.**

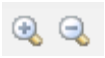


10. From the Popup window Options for FPGA Project, select Add under Configured Testbenches → select lab2_tb.vertst file from the Testbench file drop down list and click OK.



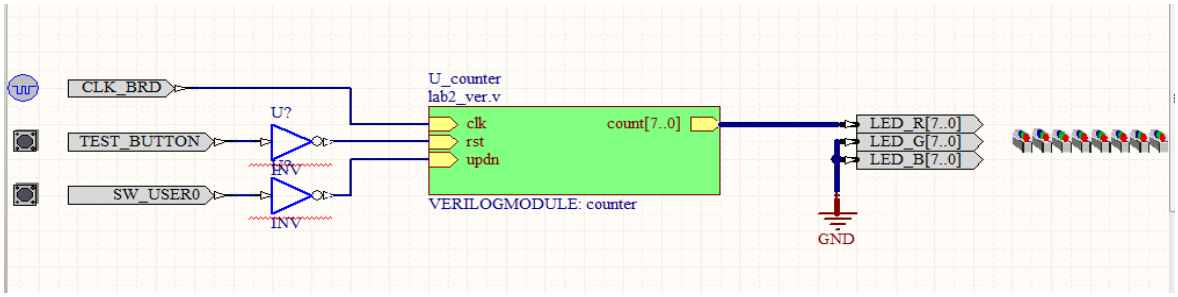
11. Select the **Options Tab** and check the option **Verilog** under **Schematic Netlister** and click **OK** to close the **options for FPGA Project** popup window**.**
12. Now select the Menu **Simulator → Simulate with Aldec OEM Simulator → Testcounter in lab2_tb.vertst.**
13. Click on **Done** in the **Edit Simulation Signals** window to proceed with the simulation using Aldec simulator.
14. From the Menu Bar select **Simulation → Run Time Step**
15. From the **Enter Time Step window** enter the **Time Step** as **1** and select the **timescale** as **ms** as shown below.



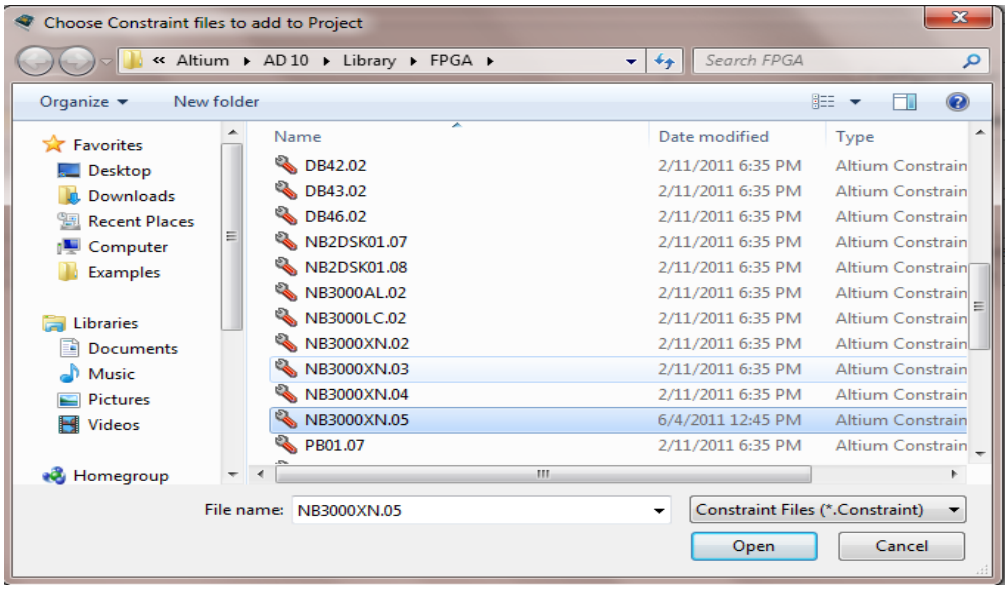
16. After successful simulation, Aldec Simulator will display the simulation results. To **Zoom In / Zoom Out** use the  icons from the toolbar. Alternatively **scroll bar** can be used by holding the **Ctrl** Key on the keypad.
17. Once the simulation results are matching with the expected results, proceed for implementing the design to FPGA

FPGA Implementation

18. Follow the below steps to create a schematic and import the HDL code as a block for implementing the design to FPGA.
19. From Projects window **RMB** on **Lab2_Prj.PrjFpg → Add New to Project → Schematic**
20. **RMB** on Schematic file and **Save** the schematic with the name **lab2_sch**
21. From the Menu bar select **Design → Create Sheet Symbol From Sheet or HDL.** From the Popup window select the file **lab2_ver.v** to generate the counter block.
22. At **RHS** select Libraries → **FPGA NB3000 Port Plugin.IntLib** and add the following components. **CLOCK_BOARD, TEST_BUTTON, USER_BUTTON0, LEDS_RGB**
23. Now select the **FPGA Generic.IntLib** from the libraries dropdown list and place the **inverter**s and wire them accordingly as shown below.



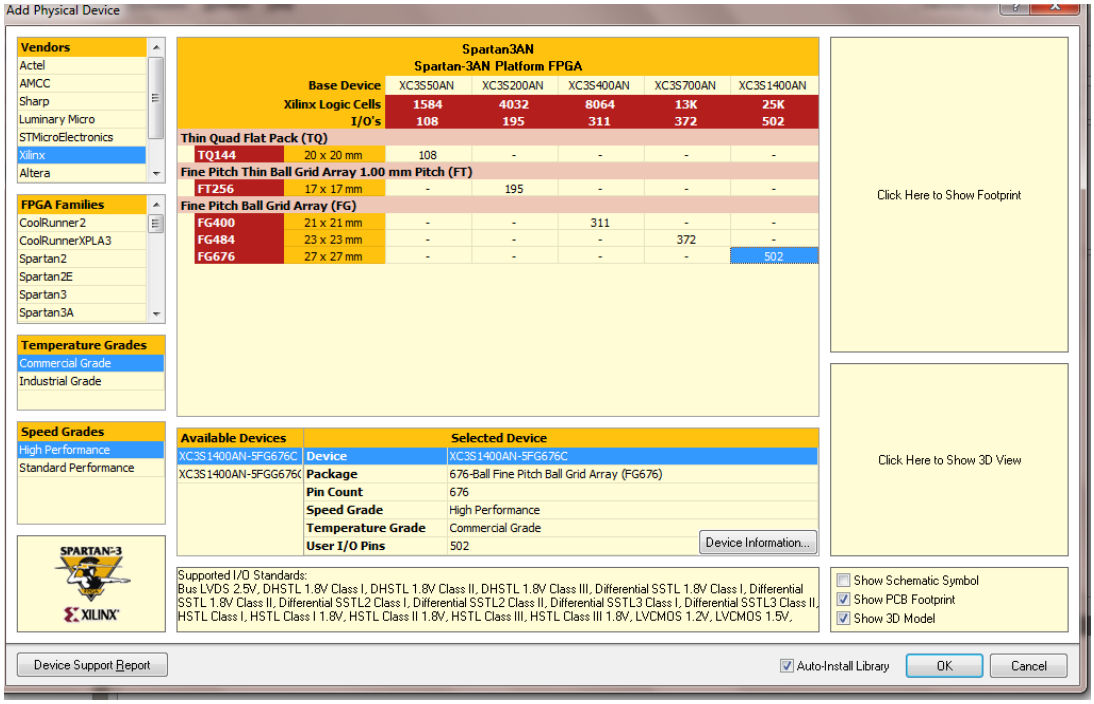
24. After Successful placement and routing, annotate the schematic designators by selecting the menu **Tools → Annotate Schematics Quietly** and Press **OK.**
25. Save the **Project** and Schematic by **RMB** on Project and Schematic file form the Projects window.
26. After saving, compile the project from Projects window by **RMB on Project → Compile FPGA Project LAB2_Prj.PrjFPG.**
27. Now **RMB** on Project from **LAB2_Prj.PrjFPG** project from Projects window and Select **Configuration Manager**.
28. Click **Add** under **Configurations** and select **Add** under **Constraint Files** option to add the constraint file.
29. From the popup menu select the file **NB3000XN.05** and check the tick mark and select OK.



30. From the Menu bar **select View → Devices View.**

31. RMB on **Virtual Device List → Add → Browse**

32. Under **Vendors** select **Xilinx**, under **FPGA Families** select **Spartan3AN**. From the Spartan3AN Family FPGA devices list at R.H.S., select **XC3S1400AN for FG676** package and click **OK** as shown below.



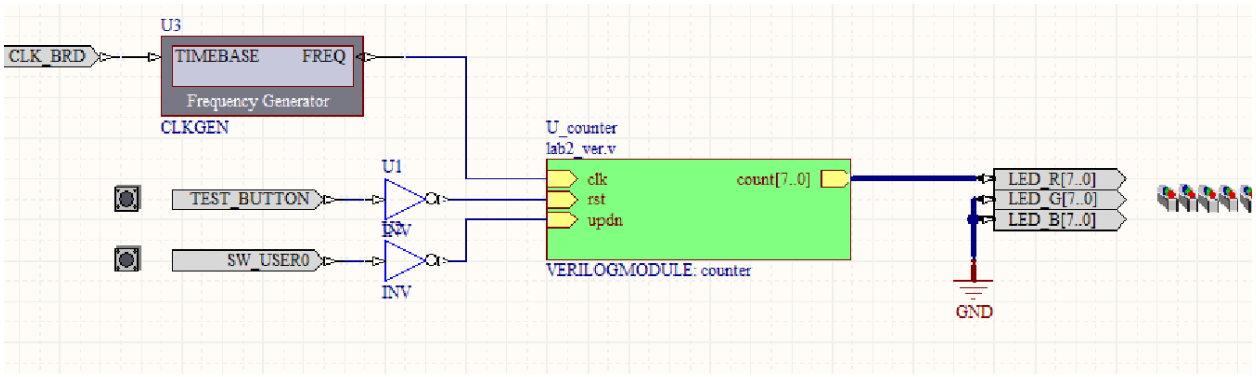
33. Click on the option Build to Compile, Synthesize and Implement the design and generate the Bit Stream for Programming the FPGA.

Connecting Altium NanoBoard 3000 to Host PC

34. **After the Build Process finishes, connect the Altium NanoBaoard 3000 to the Host PC using the USB Programming Cable. Make sure the mini USB connector at the other end of the Programming cable is connected to Host USB port beside the Power Supply Toggle Switch**.

35. Select the Option Live at the Top R.H.S of the Devices View Window to scan the Hardware Devices through JTAG connectivity. Once Altium Designer Detects the Hardware Devices on the NanoBoard, the Program FPGA option goes live. Click on the option Program FPGA to Program the Bit Stream File on to the FPGA.

36. After Programming the FPGA, LEDs will be flashing at a High Speed. To slow down the counter speed, add **CLKGEN instrument** from the **FPGA Instruments. IntLib** in the schematic as shown below and annotate the designators by selecting **Design → Annotate Schematics Quietly**.



37. Now rebuild the design from **Devices view** and **Program FPGA** with the new bit file.
38. After Programming the FPGA, scroll down to the Virtual Devices list and double click on the CLKGEN instrument.
39. Choose the desired frequency as 1 Hz / 5 Hz/ 10 Hz from the instrument rack. Now LEDs triggering can be seen on the Altium NanoBoard.

For More Documentation Visit
www.altium.com / http://wiki.altium.com/display/ADOH/Getting+Started+with+Altium+Designer