

Objectives

- Become familiar with representing and manipulating buses in VHDL
- Become familiar with bus simulations in the Xilinx ISE Simulator
- Become familiar with the numeric interpretation of buses in VHDL

Part I – The STD_LOGIC_VECTOR Type

1. Create a new project named 'slvtest' with a 10-bit bus input and one output.
2. Following Figure 1 on page 40 of your lecture notes, implement a 10-bit NAND gate (just a little different from the 10-bit AND gate shown in the figure ... you'll probably use the VHDL 'not' operator).
3. Add a new Testbench Waveform source to your project called 'slvtest_test'. In the Initial Timing and Clock Wizard, set the Combinatorial Timing Information values to 10ns for Check Outputs and 10ns for Assign Inputs.

Set the length of the test bench to $(2^{10} \cdot 20\text{ns} + 100\text{ns}) \approx 21000\text{ns}$.

4. With a bus input of type STD_LOGIC_VECTOR, the graphical simulator displays your input signal as a proper bus. To test all possible input values, click on the input waveform just past the 100ns mark (you'll have to zoom in first), then select Set Value. Click on the Pattern Wizard button. In the Pattern Type drop-down list select Count Up. Then set the Number of Cycles to $2^{10} = 1024$. Click on the Decimal radix button, then ensure that the remaining fields at the bottom of the dialog are set up to count up by 1 every clock: Initial Value is 0, Increment By is 1, Terminal Value is 1023, and Count Every is 1.

In the Processes window, you will have to right-click on Simulate Behavioral Model, select Properties, and set the Simulation Run Time to 21000ns.

Ensure that your design works properly. You may have to zoom in to see the details of each count. Also, right-click on the `inx[9:0]` bus name in the simulation results and set the radix to "Decimal (Unsigned)" to see count numbers in decimal.

5. Go back to the Implementation window and double-click on the Implement Design item in the Processes window. Then, open the Design Summary window tab and click on the Static Timing Report link (in the Detailed Reports section at the bottom).

Compare the worst-case propagation delay of this 10-input NAND gate to the 3-input AND gate from the previous learning activity (which should have been somewhere around 5 ns).

How does this correlate with your understanding of the underlying Spartan3E architecture? How many inputs does an LUT have? How many LUT's are necessary for implementing a 10-bit NAND gate?

6. Open the Implement Design tree control, then the Place & Route tree control, then double-click on the Generate Post-Place & Route Simulation Model item. Use the File->Open menu to navigate to the directory that your project files are in, then find the netgen subdirectory, and within that the par subdirectory. In that subdirectory, open up the file `slvtest_timesim.vhd` (this is the post-place & route simulation model that was generated).

Note that this is a VHDL file written in the "structural" style. It lists "primitive" components that map directly to XC3S500E resources (e.g., LUT's as component name `X_LUT4`, I/O buffers as component name `X_BUF` and `X_OBUF`, etc.) These primitive components were declared in the SIMPRIM library included at the top of the file.

Look for components of type X_LUT4. How many of them are there? See if you can follow the structural style and draw a schematic of what goes in to each LUT and what comes out.

7. Back in the Processes window, open up the Synthesize-XST tree control and double-click on the View RTL Schematic item. A new Design tab appears at the bottom of the Sources window. Click on it, open up the **slvtest** tree control and double-click on the item within. Does this schematic correspond to the post-place & route simulation model?

Remember that the “Synthesis” phase of the flow translates your VHDL code to a *generic* representation of the hardware that would implement your model. The subsequent steps of translation, map, and place&route take this generic representation and fit it onto the actual hardware resources (e.g., 4-input LUT’s) available on the device you have chosen.

8. Directly below the View RTL Schematic item you will see a View Technology Schematic item. Double-click on that, then click on **slvtest** in the Design tab of the Sources window. Now how does this schematic correspond to the VHDL post-place & route simulation model?

Part II – The UNSIGNED Type

1. Create a new project named **'addone'**, with **'addone.vhd'** as the top-level VHDL file. The goal of this circuit is to accept a 4-bit bus DATA and to output another 4-bit bus DATAOUT that is the value of the DATA bus plus one, using the unsigned interpretation of the 4-bit bus.

Make sure you leave the ENTITY declaration of the DATA and DATAOUT buses to be of type STD_LOGIC_VECTOR!

2. Use the simulator to verify that your circuit works.