Inferring a Multiplexer Demo Script

Introduction

This demonstration reviews the following procedures:

- Creating a Vivado® Design Suite project
- Creating a Source file
- Writing the code for the multiplexer
- Viewing the RTL schematic

Preparation:

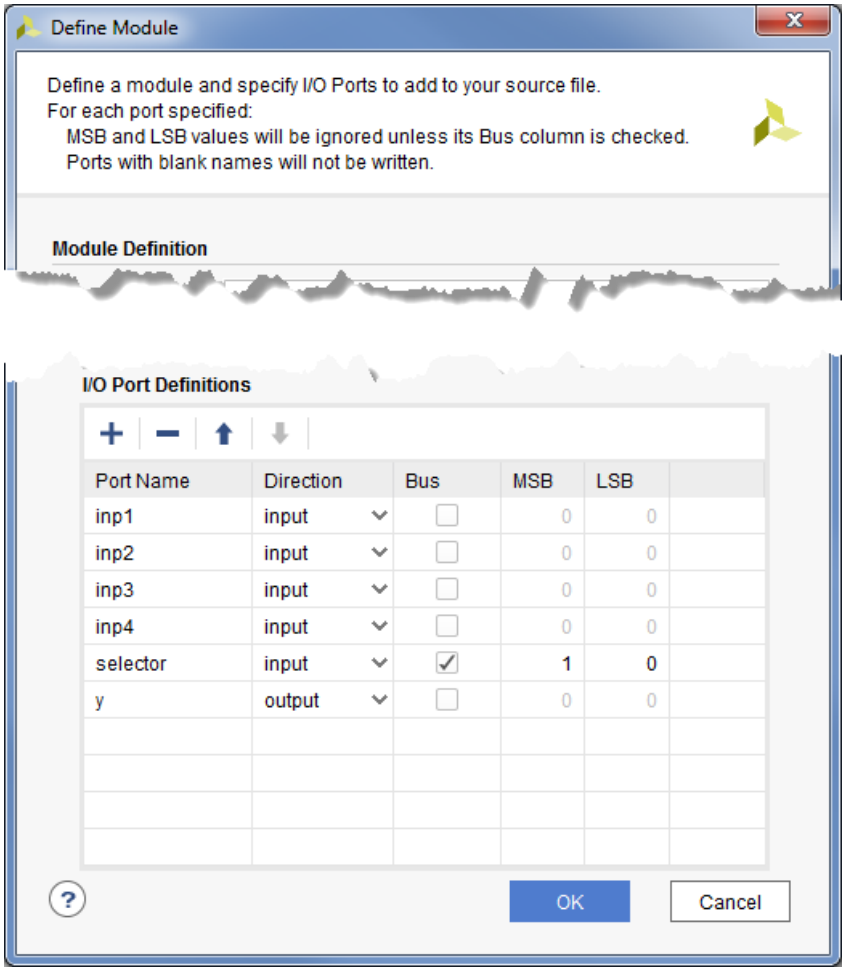
- Required files: None
- Required hardware: None
- Required software: Vivado Design or System Edition 2021.1

Inferring a Multiplexer

Action with Description	Point of Emphasis and Key Takeaway
• Launch Vivado Design Suite 2021.1.	
• Create a project using the New Project Wizard. The Create New Project Wizard in the Vivado® IDE allows designers to create a Vivado Design Suite project.	The Vivado IDE provides guidance via the Getting Started page. Here you can select from different options to start the type of project you are interested in.
• Enter **multiplexer** as the project name and choose the project location as `$TRAINING_PATH/mux/demo/KCU105`. • Enter a name for the project and specify a directory where the project data files will be stored.	The user has the ability to name things and store the project at any location.

Action with Description	Point of Emphasis and Key Takeaway
• Select **RTL Project** as the project type and select **Do not specify sources at this time**. Select the type of project to create. And since the design is based on a template, no external code is required.	Specify the type of project, which determines the types of source files that are associated with the project. Explain the different types of projects such as: • RTL project (source based) • Post-synthesis project (netlist based) • I/O planning project (part/package resources) • Imported project (from Synplify, XST, or ISE® Design Suite project file) • Example project (from a pre-defined template)
• Select the **Boards** tab and select either **Kintex-UltraScale KCU105 Evaluation Platform.** Choose a default Xilinx part or board for your project.	You can select the specific parts or an evaluation board. Filters help you quickly find your specific parts or board based on the product category, family, sub-family, package, speed grade, and temperature grade.
• Review the New Project Summary.	
• In the Flow navigator, select **Add Sources** from the Project Manager and select **Add or create design sources**. Click **Next** and select **Create File**. The file to be created is part of the design source.	The Add Sources dialog box guides you through the process of adding and creating sources for your project. Explain the different sources such as: • Constraints (defines physical parameters such as pin location) • Design sources (synthesizable) • Simulation sources (testbench)

Action with Description	Point of Emphasis and Key Takeaway
• Select the desired Language as the file type and enter **multiplexer** as the file name and select **Finish**. Create a new source file and add it to your project.	Opens the Create Source File dialog box in which you can create new VHDL, Verilog, Verilog header, memory file, or SystemVerilog files.
• Define the following ports for the multiplexer: • Port names: inp1 - input, inp2 - input, inp3 - input, inp4 - input, selector[1:0] - input (MSB - 1 LSB - 0), y - output Define a module and specify the I/O ports to add to your source file.	Explain about defining I/O ports.

Action with Description	Point of Emphasis and Key Takeaway
	
• Double-click the source file from the Hierarchy window to open the file.	Show the input/output ports assigned based on the input provided.
• Select **Tools** > **Language Templates** or **Project Manager** > **Language Templates** in the Flow Navigator. Open the language templates and refer to the multiplexer coding syntax.	Language templates provide the templates for synthesis constructs, simulation constructs, device instantiation, etc.

Action with Description	Point of Emphasis and Key Takeaway
• Expand **Verilog** > **Synthesis Constructs** > **Coding Examples** > **Multiplexers** > **Combinatorial** > **4-to-1 (always)**. OR • Expand **VHDL** > **Synthesis Constructs** > **Coding Examples** > **Multiplexers** > **4-to-1 (process)**. Copy the 4-to-1 multiplexer template.	Explain the coding examples available.
• Paste the code in the source file. • Update the code with respect to the input and output ports declared. For Verilog Users: • Change the output port **y** to reg.	

Action with Description	Point of Emphasis and Key Takeaway

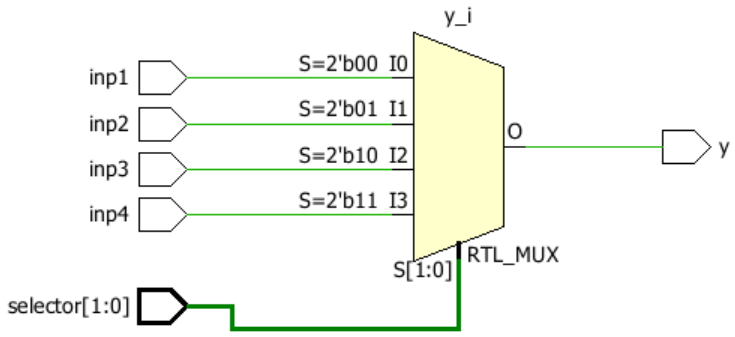
For Verilog Users:

```
23 module multiplexer(
24     input inp1,
25     input inp2,
26     input inp3,
27     input inp4,
28     input [1:0] selector,
29     output reg y
30     );
31
32    always @(selector, inp1, inp2, inp3, inp4)
33        case (selector)
34           2'b00: y = inp1;
35           2'b01: y = inp2;
36           2'b10: y = inp3;
37           2'b11: y = inp4;
38        endcase
39
40 endmodule
```

For VHDL Users:

```
43 architecture Behavioral of multiplexer is
44
45 begin
46
47 process (selector,inp1,inp2,inp3,inp4)
48 begin
49    case selector is
50       when "00" =>   y <= inp1;
51       when "01" =>   y <= inp2;
52       when "10" =>   y <= inp3;
53       when "11" =>   y <= inp4;
54       when others => y <= inp1;
55    end case;
56 end process;
57
58 end Behavioral;
```

- Save the source file.

Action with Description	Point of Emphasis and Key Takeaway
• Select **RTL Analysis** > **Open Elaborated Design** > **Schematic**. Open the elaborated design and view the RTL schematic.	The elaborated design shows how low-level logic connected to buses is represented as *grouped logic* to make the RTL schematic easier to view.
	
• Close the Vivado Design Suite.	

Summary

In this demonstration, you created a new Vivado Design Suite project, created a new Source file, accessed the language templates (the multiplexer code template), and created a multiplexer. Finally, you opened the elaborated design and viewed the RTL schematic of the multiplexer code.