Tri-State Buffer Lab

Tri-State Buffer Lab

Introduction

This lab makes use of the tri-state buffers in the XC4000XL family of devices. It will also show how to improve the performance and utilization of a 16-to-1 mux by replacing the normal CLB implementation, with tri-state buffers.

The STDMACRO project includes eight, 16-to-1 muxes contained in the macro 1621MUX. These muxes are addressed by a 4-bit binary counter. The TRIMUX project utilizes the tri-state buffers included in the XC4000E&XL architectures to implement the same muxes. The tri-state buffers are contained within the TRISTMUX macro. By using tri-state buffers, designers will yield better performance and utilization.

When first considering this design, it is easy to realize that by converting the muxes to tri-states a large number of CLBs will be saved. However, to use the tri-states, a 16-bit shift register is necessary to create a one-hot encoded counting sequence, so only one tri-state buffer will be activated at a time. This will use additional CLB registers, but provide better performance.

To complete this lab, compare the results of the STDMACRO and TRIMUX projects.

Procedure

Implementing the STDMACRO project

1) <u>Open</u> the "**Foundation Project Manager**" by <u>clicking </u>on **Start→Programs→Foundation Series→Foundation Project Manager**.

2) <u>Open</u> the **`c:\F15_labs`** directory and <u>click</u> on the "**STDMACRO**" project.

3) <u>Open</u> the "**Schematic Editor**" and <u>review</u> the project. The design should look the same as Figure 1. Use the "**Hierarchy**" button to push into any **1621MUX** macro within the design.

4) <u>Click</u> on the "**M1**" button in the Foundation Project Manager to generate an EDIF netlist and go to the M1 Design Manager.

5) To save time, this design has already been implemented for you.

6) Review the "**Place and Route**" report file by highlighting the implementation revision "`rev1(Implemented, OK)`", then clicking on **Utilities→Report Browser**, and then clicking on "**Place and Route Report**". Review this report to determine the number of CLBs used by this project, and enter this into the chart in the Questions section of the lab.

7) Open up the "**EPIC Design Editor**" by clicking on **Tools→EPIC Design Editor**. This tool will show the placement of the logic in the FPGA and assist in answering some of the questions in the "**Questions**" section. For tips on using the EPIC Design Editor, see the following section, "**Tips on Using the EPIC Design Editor**".

8) After answering question number 1, exit the **M1 Design Manager**.

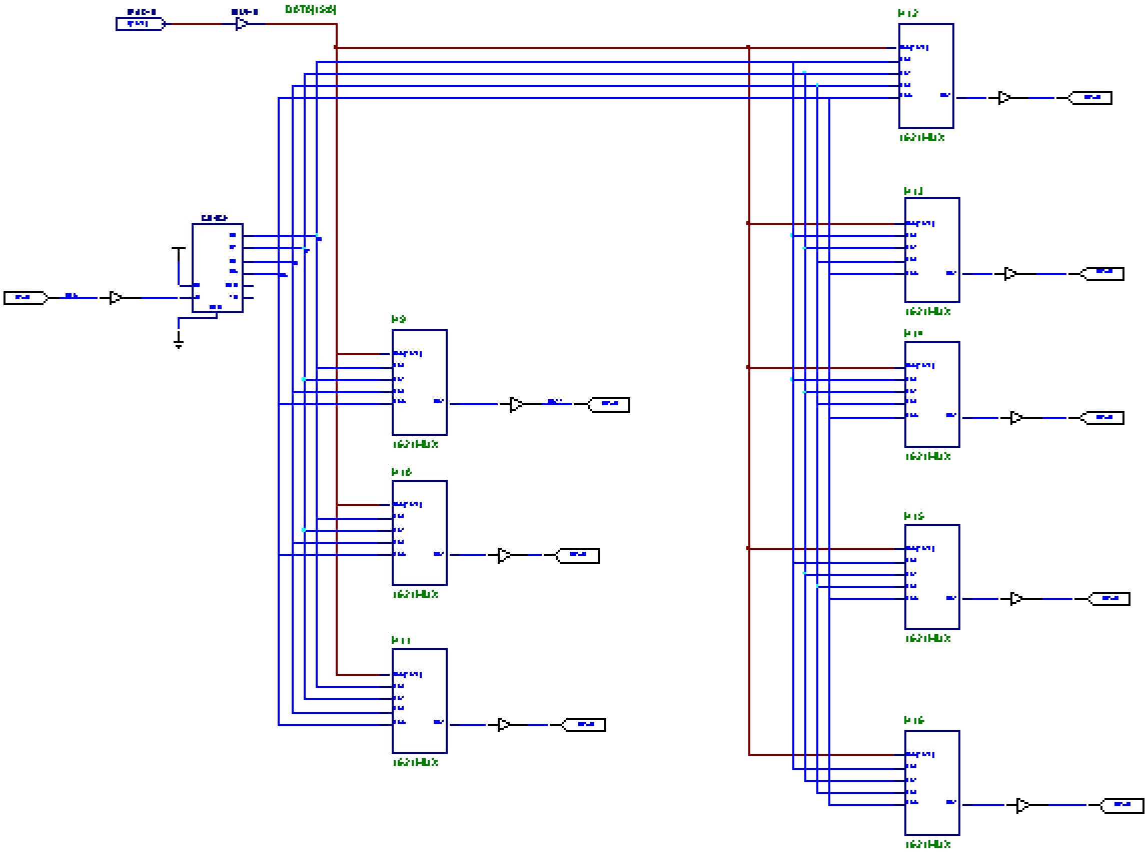
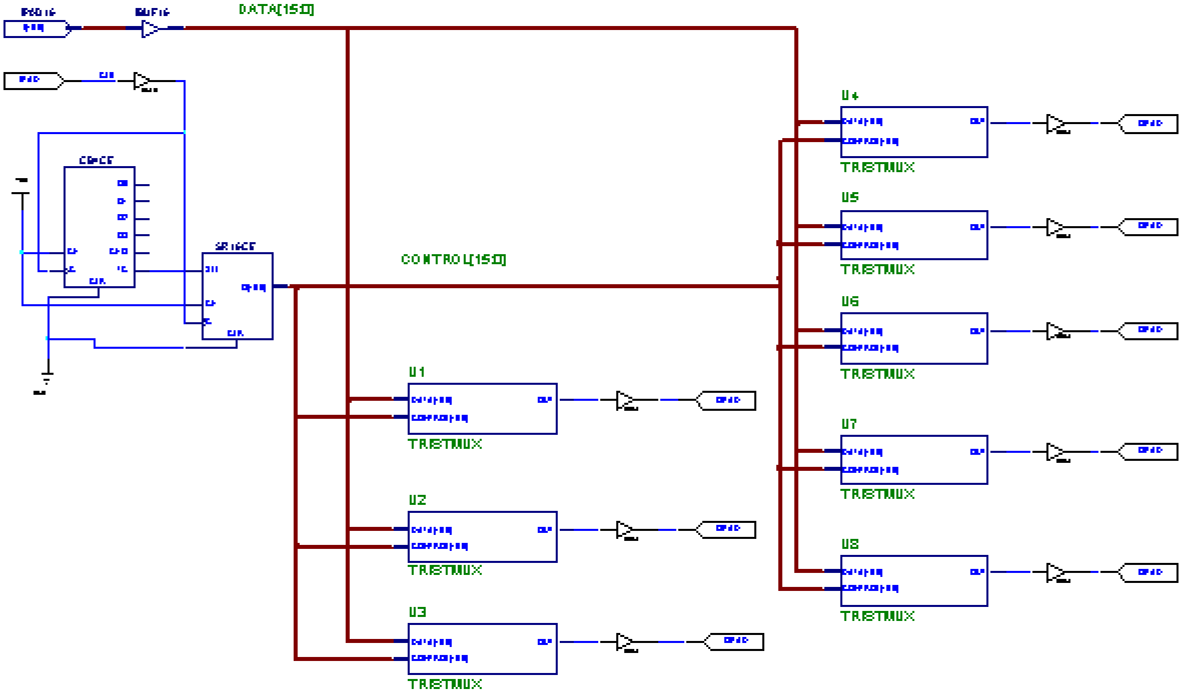


Figure 1. STDMACRO project.



Fi

Figure 2. The TRIMUX project.

Implementing the tri-state buffer version

9) Return to the "**Foundation Project Manager**" and open the "**TRIMUX**" project in the `c:\F15_labs` directory.

10) Open the "**Schematic Editor**" and review the project. It should look exactly like Figure 2. Observe the contents of the TRISTMUX macro by pushing hierarchically into the macro.

11) Click on the "**M1**" button in the Foundation Project Manager to generate an EDIF netlist and go to the M1 Design Manager.

12) To save time, this design has already been implemented for you.

13) Review the "**Place and Route**" report to complete the chart in the "**Questions**" section of this lab.

14) Open the "**EPIC Design Editor**" to help answer the remaining questions.

Tips on Using the EPIC Design Editor

The Editor for Programmable Integrated Circuits (EPIC) is a graphical application for displaying and configuring FPGAs. Although it has many sophisticated uses including enabling designers to manually place and route projects, this tool will be used to observe the chosen placement by the Flow Engine. Unless you are a very advanced Xilinx user, you will probably never need to use the EPIC tool. However, EPIC happens to be quite interesting and educational to use, as you will see!

There are five basic areas of interest in the "default" EPIC editing session.

A) **The List Dialog Box**
This box displays a list of the components, nets, layers, paths, and macros in the design. List entries are displayed in alphabetical order. Use the Option button at the top of the dialog box to specify the items to be displayed in the editing area dialog box. Select one or more items in the list with the left mouse button. When an item in the list is selected, the object in the design with that name is selected and the selected object changes color in the editing area. It is then possible to perform the same operations on the selected object that was possible when the object was selected in the editing area.

B) **The Layer Visibility Dialog Box**
The Layer Visibility dialog box allows the specification of which objects are displayed in the editing area. Select the layers to be displayed and deselect the layers that should be hidden. Select the Apply button to activate the selections.

C) **The Editing Area**
This displays a graphical representation of the interior of the FPGA device. The device components and the interconnections are displayed in this area. When editing the internal logic of a programmable component (i.e. click on a CLB and then click on "**Edit Block**") such as a logic block, a schematic of the interior of the component is displayed in a separate window.

D) **The History Area**
This area is located below the editing area. It displays commands and responses. As commands are entered from the Command Line dialog box, they are scrolled into the history and are executed. All error messages, warnings, and command responses are written into the history area. Information in this area is especially useful for deciphering unexpected command results.

E) **Locator Area**
This is displayed in the upper right corner of the EPIC window. This area shows the location of the editing area relative to the area of the entire device. As the editing area is panned and zoomed, notice the corresponding changes in the size and position of the rectangle within the locator area. Also, any objects selected appear in the locator area. The inner locator box can be dragged with the middle or right mouse button to pan the display to the desired position.

To get familiar with the EPIC Design Editor, try these steps, and then explore on your own.

15) Open up the "**EPIC Design Editor**" by clicking on **Tools→EPIC Design Editor**.

16) Select **Help → Help Topics** for the most complete help on use of the EPIC Design Editor.

17) Select **Shortcuts** for useful good tips on navigation.

18) Go back to the **EPIC Design Editor**.

19) Select **View → Zoom In** five times to zoom in on a small area.

20) Under **Layer Visibility**, check **Long Lines** and click on **Apply**. Now you can see all of the available long line resources.

21) Select **Tools → Find…**, type "**Q2**" under the **Name:** option, and click <**OK**>. The viewer will highlight and pan to the block containing the component named Q2. This is one of the flip flops from the counter in the original schematic design.

22) Select **Edit → Edit Block** to see how this block was implemented with the CLB resources.

23) Click **Cancel** to exit the Edit Block mode.

24) In the **EPIC List** window, select the **CLK** net. Notice that the viewer highlights the CLK net through the chip in both the world view and the main view windows.

25) Experiment further on your own.

Xilinx also offers a free graphical **Floorplanner** tool. The Floorplanner is a standard included tool with version 1.5 software.

QUESTIONS

Mux Implementation	**Number of CLBs Used**
STDMACRO	
TRIMUX	

1) Describe the CLB placement found in EPIC for the STDMACRO design.

2) Describe the CLB and tri-state buffer placement found in EPIC for the TRIMUX design.

3) What is unusual about the orientation of the tri-state buffers in the TRIMUX design?

   What is the reason for this?

4) What is unusual about the orientation of the CLBs in the TRIMUX design?

   What is the reason for this?

5) Why was each of these labs implemented in an XC4005XL, rather than an XC4002XL when only a few CLBs are used?

6) Try reimplementing the TRIMUX project in an XC4002XL-3. To try this, bring up either project in M1 and after clicking on the "**start**" button, select an XC4002XL-3 device. What error message is seen?

Conclusion

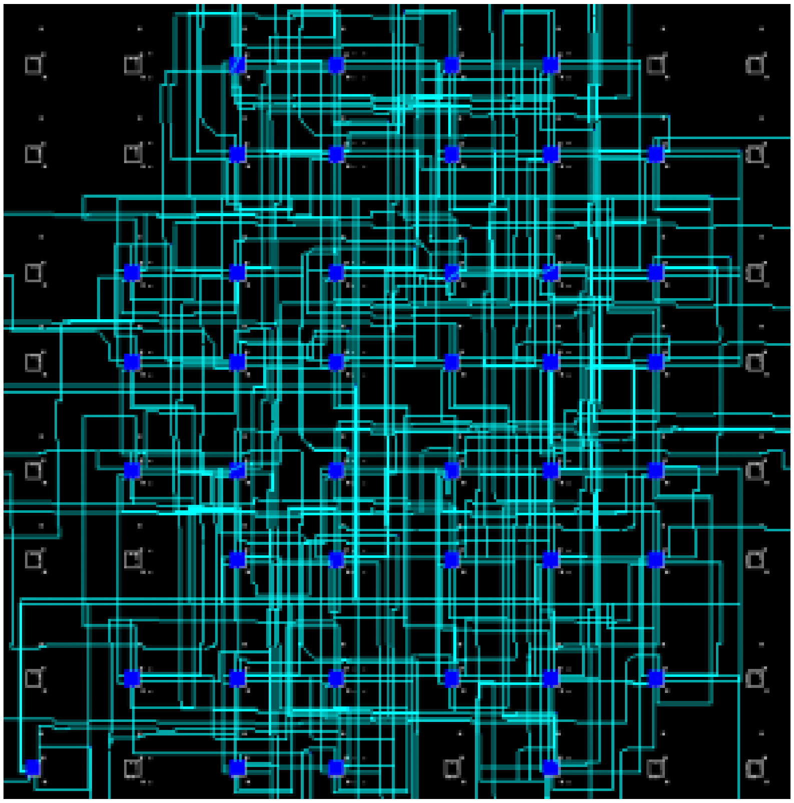
From this lab, it is easy to see how to use tri-state buffers in a design. They are especially useful for Muxes and Wired-and gates. Tri-states allow more logic to fit into the XC4000 devices by saving CLBs. Tri-state buffers are also very fast and can provide outstanding performance, while enabling higher utilization.

ANSWERS

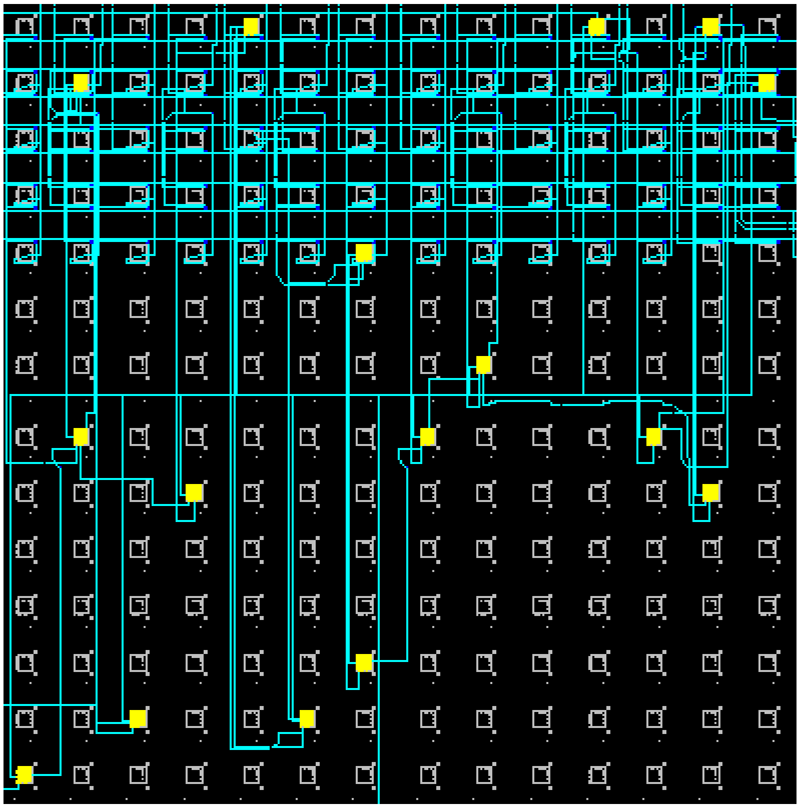
Here are the results found after implementing all the designs. (Remember that the results may vary).

Mux Implementation	Number of CLBs Used
STDMACRO	~ 42 CLBs
TRIMUX	~ 18 CLBs

1) This is what was found for an initial placement recommended by M1 for the STDMACRO design. Your results may vary. Note how many CLBs are used.



2) This is what was found for an initial placement recommended by M1 for the TRIMUX design. Your results may vary.



3) All the tri-state buffers are in the same row so they can have their outputs connected to the same longline. Every tri-state buffer connects to a single horizontal longline, and every tri-state buffer in a row drives the same longline.

4) The CLBs are placed throughout the device to give the design the most flexibility to route additional logic that may be put into the project. If there had been a timing specification on the logic, the CLBs would have probably been placed closer together.

5) These labs were implemented in an 4005XL-3 because there are fourteen CLBs in each row, with a tri-state buffer above and below each CLB. Since each of the two tri-state buffers is connected to different horizontal longlines, each horizontal longline can connect the outputs of fourteen tri-state buffers. However, there are an additional two tri-states that can connect to each longline that are associated with each IOB at the end of each longline. Therefore, since the XC4002XL has only 8 CLBs in each row, it can only connect up to ten tri-states to each longline. Since we are replacing eight, 16-to-1 muxes with tri-states, an XC4005XL-3 device is needed.

   For additional information on tri-state buffer architecture in the XC4000XL or E families, refer to the Xilinx Data Book.

6) The error message "Too many TBUFs…to fit on one row…" is seen.