## Lab **Multiplexers** Class Date Name **Objectives** Upon completion of this laboratory exercise, you should be able to: • Enter the logic circuit of a 4-to-1 multiplexer (MUX) as a Block Diagram File, using Altera's Quartus II CPLD design software. • Create a Quartus II simulation file for the 4-to-1 multiplexer described above. • Create a hierarchical design in the Quartus II Block Editor that contains a multiplexer and other components. • Download the 4-to-1 MUX to a CPLD test board and test its function. Reference Dueck, Robert K., Digital Design with CPLD Applications and VHDL, 2/e Chapter 6: **Combinational Logic Functions** 6.3 **Multiplexers** Equipment **CPLD** Trainer: Required Altera UP-2 board with ByteBlaster Download cable, or DeVry eSOC board, with parallel port cable, or RSR PLDT-2 board with Straight-Through Parallel Port cable, or equivalent CPLD trainer board with Altera EPM7128S CPLD Quartus II Web Edition software AC adapter, minimum output: 7 VDC, 250 mA DC Anti-static wrist strap #22 solid-core wire Wire strippers Oscilloscope

# **Experimental Notes**

A multiplexer (abbreviated MUX) is a device for switching one of several digital signals to an output, under the control of another set of binary inputs. The inputs to be switched are called the **data inputs;** those that determine which signal is directed to the output are called the **select inputs.** 

Figure 13.1 shows the logic diagram of a 4-to-1 multiplexer, with data inputs labeled  $D_3$  to  $D_0$  and the select inputs labeled  $S_1$  and  $S_0$ . By examining the circuit, we can see that the 4-to-1 MUX is described by the following Boolean equation:

$$Y = D_0 \overline{S}_1 \overline{S}_0 + D_1 \overline{S}_1 S_0 + D_2 S_1 \overline{S}_0 + D_3 S_1 S_0$$

For any given combination of  $S_1S_0$ , only one of the four product terms will be enabled. For example, when  $S_1S_0 = 10$ , the equation evaluates to:

$$Y = (D_0 \cdot 0) + (D_1 \cdot 0) + (D_2 \cdot 1) + (D_3 \cdot 0) = D_2$$



Figure 13.1 4-to-1 Multiplexer

The MUX equation can be described by a truth table, as in Table 13.1. The subscript of the selected data input is the decimal equivalent of the binary combination  $S_1S_0$ .

Multiplexers can be implemented in Quartus II as a Block Diagram File similar to Figure 13.1 or as a VHDL design entity. Implementation and applications of multiplexers are described in more detail in *Digital Design with CPLD Applications and VHDL, 2/e* on pages 297–313.

| Table 13.1            | Trut<br>a 4-t         | h Table of<br>to-1 MUX |
|-----------------------|-----------------------|------------------------|
| <b>S</b> <sub>1</sub> | <i>S</i> <sub>0</sub> | Y                      |
| 0                     | 0                     | $D_0$                  |
| 0                     | 1                     | $D_1$                  |
| 1                     | 0                     | $D_2$                  |
| 1                     | 1                     | $D_3$                  |

# Procedure

### Block Diagram File and Simulation for 4-to-1 Multiplexer

Create a Block Diagram File for a 4-to-1 multiplexer as shown in Figure 13.1. Save the file as *drive*:\qdesigns\labs\lab13\4to1mux\4to1mux.bdf (Tip: You can place the inverters vertically, as shown in Figure 13.1, by entering the NOT symbol, right-clicking on the symbol, and choosing Rotate, 270° from the pop-up menu.)

Compile the design.

2. We can use the following set of simulation criteria to test the 4-to-1 multiplexer.

#### Simulation Criteria

- Each data input channel of the multiplexer will be selected in an ascending sequence by applying a binary count to the combined select inputs.
- Each data input should be easily recognizable by having a "signature" waveform applied to it. Each channel should be selected for a period no less than about two or three cycles of the signature waveform.
- The output waveform should display a series of unique signature waveforms, indicating the selection of the data channels in the correct sequence.

**3.** A set of signature waveforms can be created by applying waveforms of different frequencies that are related by increasing binary multipliers. If, for example, we have a count waveform with a base period of 20 ns, we can create other waveforms with periods of 40 ns, 80 ns, and 160 ns, which represent multipliers of 2×, 4×, and 8× the base period. In addition, we should choose a period for the select inputs such that we see about three cycles of the slowest waveform, in this case 960 ns. This set of waveforms is shown in Figure 13.2. The waveforms are staggered in their frequencies so as to make the largest contrast between adjacent waveforms.

| \$  | 4to                  | o1mux Simulat | ion Repor | rt 🛛  | - 🗆 ×   |  |  |
|-----|----------------------|---------------|-----------|---|---------|--|--|
| l s | Simulation Waveforms |               |           |   |         |  |  |
| M   | /last                | er Time Bar:  | 0 ps      | Pointer: 5.81 ns Interval: 5.81 ns Start: End:  |         |  |  |
| IΓ  |                      |               | Value at  | 0 ps 320,0 ns 640,0 ns 960,0 ns 1.28 us 1.6 us 1.92 us 2.24 us 2.56 us 2.88 us 3.2 us 3.52 us | 3.84 us |  |  |
|     |                      | Name          | 0 ps      |   |         |  |  |
| lĒ  |                      | 🔳 s[10]       | B 00      |   |         |  |  |
|     |                      | d0            | BO        |   |         |  |  |
|     |                      | d1            | BO        |   | inininr |  |  |
|     |                      | d2            | BO        |   |         |  |  |
|     |                      | d3            | BO        |   | υw      |  |  |
|     |                      | у             | BO        |   | n n     |  |  |
| IΓ  |                      |               |           |   |         |  |  |

Figure 13.2 Simulation for a 4-to-1 Multiplexer

4. Create the simulation waveforms of Figure 13.2, using an end time of 4  $\mu$ s. Apply a count value to each of the inputs using the count value specified in Table 13.2.

To apply a count value to a waveform, highlight the waveform and click the **Count Value** button on the Waveform Editor toolbar, as shown in Figure 13.3. In the **Count Value** dialog box, select the **Timing** tab and fill in the value for **Count every:** as shown in Table 13.2.

| Fable 13.2SimulationSpecifications |              |
|------------------------------------|--------------|
| Input                              | Count every: |
| s[20]                              | 960 ns       |
| d0                                 | 160 ns       |
| d1                                 | 20 ns        |
| d2                                 | 80 ns        |
| d3                                 | 40 ns        |



**Figure 13.3** Waveform Editor Toolbar

5. When you have completed the input waveforms, run the simulation and show your instructor.

### Testing the 4-to-1 MUX

One way to test the MUX function on the CPLD board is to apply a known signal to each data input, as we did in our simulation, and manually change the values of the select inputs with DIP switches. The output signal can be observed on a monitoring device such as an LED or an oscilloscope. The LED or oscilloscope will tell you which MUX input channel has been selected.

Two MUX test circuits are shown in Figures 13.4 and 13.5. The test circuits are examples of **hierarchical design.** All this means is that the test circuit (called the **top level** of the hierarchy) contains components that are complete designs in and of themselves. The design hierarchy shown contains two components: a 4-to-1 multiplexer, created from the block diagram file in



Figure 13.4 Test Circuit for a 4-to-1 MUX (Altera UP-2)



Figure 13.5 Test Circuit for a 4-to-1 MUX (RSR PLDT-2 or DeVry eSOC)

Figure 13.1, and a predesigned clock divider, **clkdiv1** or **clkdiv1a** for the Altera UP-2 board or **clkdiv2** or **clkdiv2a** for the RSR PLDT-2 or DeVry eSOC board.

The clock dividers in Figures 13.4 and 13.5 (**clkdiv1** (Altera UP-2) or **clkdiv2** (RSR PLDT-2 or DeVry eSOC) provide digital square wave signals of binary-multiple frequencies at four different outputs (Altera UP-2: 1.5 Hz, 3 Hz, 6Hz, and 12 Hz; RSR PLDT-2 and DeVry eSOC: 1 Hz, 2 Hz, 4 Hz, 8 Hz). These frequencies are slow enough to be observed visually.

An alternate pair of clock dividers, **clkdiv1a** (Altera UP-2) and **clkdiv2a** (RSR PLDT-2 and DeVry eSOC) are used to view the output waveforms on an oscilloscope. **Clkdiv1a** has a base MSB frequency of about 1.5 kHz, derived from the 25.175 MHz oscillator on the Altera UP-2 board. **Clkdiv2a** has a base MSB frequency of about 1 kHz, derived from the 4 MHz oscillator on the RSR PLDT-2 or DeVry eSOC board.

To create the test circuit, we must first create the symbols for the components, as follows.

1. Create a new folder for the project and call it:

#### drive:\qdesigns\labs\lab13\test4to1

- 2. Copy the files 4to1mux.bdf and clkdiv1.vhd to the new folder.
- Create symbols for the MUX and clock divider as follows. Open the file 4to1mux.bdf in Quartus II. From the File menu, select Create/Update, Create Symbol Files for Current File. Repeat this procedure for the file clkdiv1.vhd
- 4. From the File menu, shown in Figure 13.6, select New Project Wizard.

| D  | <u>N</u> ew                 | Ctrl+N             |
|----|-----------------------------|--------------------|
| Ê  | Open                        | Ctrl+O             |
|    | ⊆lose                       | Ctrl+F4            |
| 湓  | New Project <u>W</u> izaro  | J                  |
| ŝ  | Open Project                |                    |
|    | Convert MAX+PLU             | 5 II Project       |
|    | Clos <u>e</u> Project       |                    |
|    | Save                        | Ctrl+5             |
|    | Save <u>A</u> s             |                    |
|    | Save Current Repo           | rt Section As      |
| ø  | Save All                    |                    |
|    | Eile Properties             |                    |
|    | Create / Update             | •                  |
|    | Export                      |                    |
|    | Convert Programm            | ing Files          |
|    | Inclu <u>d</u> e Report Sec | tion in Print List |
| 10 | Page Setyp                  |                    |
| B. | Print Preview               |                    |
| 6  | Print                       | Ctrl+P             |
|    | Recent Files                | •                  |
|    | Recen <u>t</u> Projects     | ¥                  |
|    | Exit                        | Alt+F4             |

Figure 13.6 New Project Wizard (File Menu)

| www.project.wizard: Directory, Name, and Top-Level Entity   | ge 1 of 6] 🛛 🗙                           |
|---|--|
| What is the working directory for this project? This directory will co<br>other related files associated with this project. If you type a directo<br>exist, Quartus II can create it for you. | in design files and<br>ame that does not |
| C:\qdesigns\Labs\Lab13\test4to1   |  |
| What is the name of this project? If you wish, you can use the nar<br>top-level design entity.  | of the project's                         |
| test4to1  |  |
| in this wizard. After you create a project, you can add more top-lev<br>Compiler and Simulator settings for them with commands on the As  | entities and create<br>iments menu.      |
| test4to1  |  |
|   |  |
|   |  |
|   |  |
|   |  |
|   |  |

Figure 13.7 New Project Wizard (Directories)

- **5.** In the first screen of the New Project Wizard, browse to select the new folder as shown in Figure 13.7.
- 6. In the second screen of the New Project Wizard, click Add All to get the screen shown in Figure 13.8.
- 7. Continue through the screens of the New Project Wizard to select the Device Family (MAX 7000S) and target device (EPM7128SLC84-7, -10, or -15).
- **8.** Click the **New File** icon or select **New** from the **File** menu. From the resultant dialog box, shown in Figure 13.9, select **Block Diagram/Schematic File** and click **OK**.
- Double-click on the Block Editor desktop to get the Symbol dialog box, as shown in Figure 13.10. Enter the symbol for the file 4to1mux, as found in the Project folder of the Symbol dialog box. Repeat the procedure for the clock divider component.

Add input and output pins, as shown in Figure 13.4 or 13.5.

Create an output for disabling unused LEDs: **unused [6..16].** For the Altera UP-2 board, connect these LED outputs to  $V_{CC}$ . For the RSR PLDT-2 board, connect the LED outputs to ground. Draw the connecting line from the output pin symbol to the VCC or GND symbol, not the other way around. Make sure that the connecting line is a thick line, indicating a connection to multiple output lines for one pin symbol. The line thickness can also be changed by highlighting the line, then right-clicking it and choosing Bus Line from the pop-up menu. Save and compile the file. The **unused** output is not required for the DeVry eSOC board.

| ile name:                  |                                   | ]                  | Add        |
|----------------------------|-----------------------------------|--------------------|------------|
| File name                  |                                   | Туре               | Add All    |
| 4to1mux.bdf<br>clkdiv1.vhd |                                   | VHDL File          | Remove     |
|                            |                                   |                    | Properties |
|                            |                                   |                    | Up         |
|                            |                                   |                    | Down       |
| your project incl          | udes libraries of custom function | ons, specify their |            |

Figure 13.8 New Project Wizard (Add Files)

| AHDL File   |                |   |  |
|---|----------------|---|--|
| Block Utagra<br>EDIF File<br>Verilog HDL I<br>VHDL File | m/Schematic Fi | e |  |
|   |                |   |  |
|   |                |   |  |

Figure 13.9 New File Dialog Box (Block Diagram File)



Figure 13.10 Symbol Dialog Box

**10.** Once the design hierarchy has been created, the component files can be accessed for editing by double-clicking. Try double-clicking on the component called **4to1mux**.

What do you see? \_\_\_\_\_

- **11.** Assign pin numbers as shown in Table 13.3. When you have assigned the pin numbers, save and compile the file again.
- 12. Connect short lengths of #22 solid-core wire from the prototyping headers around the EPM7128S chip to two DIP switches (for  $S_1$  and  $S_0$ ) and the LEDs, as required.

On the RSR board, if no wire connections are in place, it is not necessary to do any wiring. On the Altera UP-2 board, if the wire connections are not in place, it is only necessary to connect  $S_1$ ,  $S_0$ ,  $d_0$ ,  $d_1$ ,  $d_2$ ,  $d_3$ , and Y. The CLOCK connection is hardwired on both boards.

The pin assignments for the unused LEDs are there only so that these LEDs are not lit if they have been wired for another project, so don't connect them if they are not already connected.

13. Download the test circuit to the CPLD test board. Set the  $S_1$ ,  $S_0$  switches to 00 and observe the output of the MUX on the output LED. Repeat for values of 01, 10, and 11. Explain your observations to your instructor. Compare the **y** output to the MUX inputs displayed on the **d** LEDS.

Instructor's Initials

| Pin Number |      |        |      |
|------------|------|--------|------|
| Pin Name   | UP-2 | PLDT-2 | eSOC |
| s1         | 34   | 34     | 50   |
| sO         | 33   | 33     | 51   |
| clock      | 83   | 83     | 83   |
| У          | 52   | 52     | 15   |
| d0         | 44   | 44     | 4    |
| d1         | 45   | 45     | 5    |
| d2         | 46   | 46     | 8    |
| d3         | 48   | 48     | 9    |
| unused[6]  | 49   | 49     | 10   |
| unused[7]  | 50   | 50     | 11   |
| unused[8]  | 51   | 51     | 12   |
| unused[9]  | 80   | 80     | 16   |
| unused[10] | 81   | 81     | 17   |
| unused[11] | 4    | 4      | 18   |
| unused[12] | 5    | 5      | 20   |
| unused[13] | 6    | 6      | 21   |
| unused[14] | 8    | 8      | 22   |
| unused[15] | 9    | 9      | 24   |
| unused[16] | 10   | 10     | 25   |

 Table 13.3
 Pin Assignments for 4-to-1 MUX Test Circuit

## Monitoring the MUX Output with an Oscilloscope

- 1. Modify the test circuit in Figure 13.4 or 13.5 so that the clock divider is replaced by either clkdiv1a or clkdiv2a. (You will have to create the component first.) Compile the project again.
- 2. Program your CPLD test board with the new circuit. With an oscilloscope, monitor the waveforms at the MUX inputs, one at a time, and measure the frequency of each one. Fill in Table 13.4 with the results of your measurements.

| Table 13.4 Fr<br>W | Frequencies of<br>Waveforms Applied<br>to MUX Inputs |  |
|--------------------|--|--|
| Pin name           | Frequency  |  |
| d0                 |  |  |
| d1                 |  |  |
| d2                 |  |  |
| d3                 |  |  |

**3.** Monitor the output y with the oscilloscope. Change the values of  $S_1$  and  $S_0$  to get all combinations of these inputs. For each combination, fill in Table 13.5 with the frequency of the waveform at output y and which selected input waveform this corresponds to.

| Table 13              | .5 Freque      | requency of MUX Output with Different Selected Inputs |                |  |
|-----------------------|----------------|---|----------------|--|
| <i>S</i> <sub>1</sub> | S <sub>0</sub> | Frequency at output y                                 | Selected input |  |
| 0                     | 0              |   |                |  |
| 0                     | 1              |   |                |  |
| 1                     | 0              |   |                |  |
| 1                     | 1              |   |                |  |

T | | 40 - F . . . . . . .