Using the 3700 library parts in ISE schematics

Your lab kit includes a variety of 7400-series chips, switches, LEDs and other components. Those components are described in the lab kit handout on the class web site (www.eng.utah.edu/~cs3700). Some of those components (mostly the gates) can use Xilinx gates as stand-ins for the 7400-series gates. But, there are no Xilinx versions of the red switches, no equivalent of the 907 and 906 open-drain drivers, and no LEDs in the Xilinx symbol libraries.

The 3700 library does have these parts so you can make a schematic that includes these components using the ISE schematic editor. It used to be that you could attach this library the same way that you use the attached library of Spartan2 parts (i.e. the gates). This may still work at some point (I have a support case pending at Xilinx about this). But, for now, it doesn’t work that easily so we’ll have to use a workaround.

The workaround is for you to get the 3700_parts library from the class web site, put it on your machine somewhere, and import the parts from that project into your own Lab1 project so you can use them. Here are the steps:

1. Download the 3700_parts.zip file from the class web site and unpack it. The resulting 3700_parts folder contains the project that has all the 3700-specific parts in it. I put mine on the desktop, but you can put it wherever you like.

2. Make a new project for Lab1. I called mine Lab1 (not very original, but it does get the point across). Follow the steps in the tutorial and make sure you use the correct spartan2 part for the library. It’s not critical for this lab, but it’s a good habit to get into.

3. In the Project menu, select “Add Copy of Source”
4. Navigate to the 3700Parts folder, and change the “Files of Type:” field to “All Files:” so you can see them all:
5. Select all the parts with type .sch and .sym to open:

6. You should see green checks on all the files in the “Adding Source Files” dialog box.
7. After clicking OK you should see those components in your Sources list. You can now use them in your schematics along side the Xilinx components.

```
Sources for: implementation
- Lab1
  - xc2s50-6q144
    - LED_906 (LED_906.sch)
    - LED_907 (LED_907.sch)
    - SPDT (SPDT.sch)
    - SPST (SPST.sch)
    - buf_906 (buf_906.sch)
    - buf_907 (buf_907.sch)
```

8. For example, I can use a SPST switch (the read switches are actually SPDT but I can only wire up one throw if I want to use it as a SPST), pullup and gnd (from the Xilinx "General" category), a 907 buffer, and a 907-LED to make a simple schematic of a circuit where the LED goes on when I flip the switch. It looks like this:

You should be able to see the function of this circuit, and how to wire it up with the components from the lab kit, by looking carefully at the Lab Kit handout (see figures 2, 3, 4, and 5, for example).
9. Making a new test fixture named switch_tf, and making sure to associate it with the switch.sch schematic, I get the following test fixture template:

```
// Verilog test fixture created from:

`timescale 1ns / 1ps

module switch_switch_sch_tb();

    // Inputs
    reg Switch;

    // Output
    wire Light;

    // Bidirs

    // Instantiate the UUT
    switch UUT |
    .Light(Light),
    .Switch(Switch)

    );

    // Initialize Inputs
    `ifdef auto_init
    initial begin
    Switch = 0;
    `endif

endmodule
```

10. I can fill this in with an initial block to drive the input (switch) and look at the output (light). If I got it right, the light should light up whenever I flip the switch value (which simulates actually flipping the switch).
11. The simulation shows that it works. Note that in order to make this intuitive to use, the interpretation of the switch “S” input is a little tricky. When S is 0, the switch is closed (so the output of the switch is pulled low). The S is 1 the switch is open, so the output of the switch is pulled high through the pullup. This lets you drive the S signal with the intended output of the switch.

The simulation shows how this works. When my Switch input is low, the switch is closed so the input to the buffer is low (pulled low through the switch) and the light is off (low). When my Switch input is high, the switch is open, so the input to the buffer is pulled high through the pullup and the light is on (high)
This example has showed how to add the cells in the 3700Parts library to your Lab1 project so that you can use schematic symbols for the red switches, the pullup resistors, the 907 and 906 drivers, and the LEDs in the lab kits. Use these symbols, along with the Xilinx nand and inv parts, to build your three-way switch circuit for Lab1.