

Lab # 4

Macros

Objectives

- Familiarization with circuit macros.
- More familiarization with FPGA board and design tools.
- To build a set of I/O macros for future reference
- To add I/O macros in the library of modules so that they can be used in the future.

Description

This lab requires you to

- Build *macro of* the schematic of the last lab.
- Build macros for input switches and buttons
- Build macros for output LEDs.
- Build a library of input switches and push buttons as well as the output LEDs.
- Functional simulation and implementation of the circuit by replacing the repeating sub-circuits and the input/output terminals with macros.
- Verifying all the I/O macros by connecting an input macro to an output one.

Creating Macros

Often, you will use a circuit (consisting of different logic gates) again and again, for example, a seven-segment decoder or a full adder. Instead of drawing the full logic again, it would be more efficient to make a module or macro of it with its own symbol. This macro can then be added to library and can be used like other available devices. A macro hides all the underlying logic and simplifies the design by providing only the interface signals for use in the design. The macro then gets added to your symbol library. You can use it like you use ordinary symbols of the library.

A macro's creation is essentially the same as creating the schematic with just two main differences.

Since the output of the macro need not be routed outside the FPGA, you don't need to add input/output buffers (IBUF, OBUF).

You do not use IPADs or OPADs for the input and output ports. Use I/O terminals instead to indicate the input and output I/Os (More on this later).

A macro can be created by three different methods

Using Symbol Wizard

By using this method, the wizard automatically adds I/O terminals and adds the symbol in the library for you. What remains is to draw the logic of the macro.

From an Existing Schematic

You might face a situation where you want to convert an existing schematic into macro. Here you will need to manually replace the I/O Pads and Buffers with I/O terminals and add the macro to the symbol library.

Creating Schematic First

This is essentially the same as creating macro from the schematic. The only difference is instead of replacing I/O Pads and Buffers from the schematic with I/O terminals, you add I/O terminals directly.

Using Symbol Wizard

Open the Schematic Entry tool by clicking on the Schematic Entry icon in the Project Manager Window.

In the Schematic Entry window, go to the **Tools\Symbol Wizard**. A Wizard will guide you through by asking several questions.

- For the Symbol Name, type in the name you want to give, e.g. 7Decoder (for seven segment decoder). Select the Schematic for Contents.
- Then click the NEXT button. Specify, by filling in, the names of the input and output ports. When all input and output ports have been defined, click the NEXT button. Follow the instructions.
- When finished click the FINISH button.

The symbol will now be created and added to the project library. An underlying design file will also be created that has the defined port names. The schematic editor will open showing an empty schematic page for the macro, except for the input and output pins. Then, you draw the logic for the macro in the same way as you draw a new schematic.

Using an existing schematic as a macro

In case you created a schematic as part of another project and would like to use this schematic as a macro in a new project you can do so. The steps are as follows:

- First you will have to add the old schematic to the current (new) project. In the Project Manager window, go to the **File\Add Document** menu. Use the browser to locate the schematic of interest. When done, you will see that the schematic has been added to the project in the left window page (Files tab). Let's assume that the schematic is named TEST1.sch.
- You may want to modify this schematic, e.g., to change the pin names, replace input PADS (IPADS) by I/O terminals and remove buffers (BUF). Remember that for macros you do not use IPADS or OPADS (these indicate the physical pins of the device) but I/O terminals
- Next you need to create a macro from the schematic. This is done as follows: go to **Hierarchy\Create Macro Symbol from Current Sheet**. A window will pop up. You can change the name of the macro, and add a brief description in the comment line. Let's call the macro MACRO1. You can also check the input and output signals. Click OK.
- When you convert the schematic **TEST1.sch** into the macro MACRO1, there is no longer any relationship between the schematic TEST1.sch and the project, even though you can still open the schematic TEST1.sch. You can remove the schematic from the project directory using the Windows Explorer.

I/O Terminals

You can add I/O terminals (pins) by clicking on the I/O Terminal icon (hierarchy connector) on the left-hand side of the toolbar or at the top of the SC Symbols window. A pop-up Terminal window will open. Fill out the name of the terminal (ex. X, Y or Z) and indicate whether it is an Input or Output port. Be careful to indicate the right direction for the inputs and outputs.

Managing Macros

Often you might be required to add a macro that you developed previously. The macro is saved internally in the library of the project in which it was developed. If that macro is required, the library of that project is to be added in the current project libraries. This is done using Project Libraries (File/Project Libraries...) window (Figure 2.16).

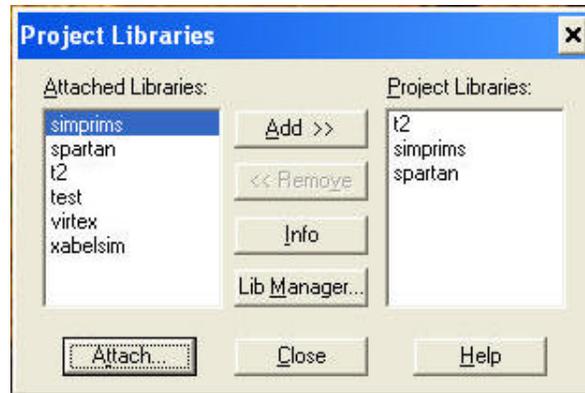


Figure 2.16: Adding the Library

If the previous library is not available in left hand list of libraries (Attached Libraries), the library would be first attached using Lib Manager.

In the library Manager window, attach the library by invoking Attach Library dialog box (Library/Attach...) as shown in Figure 2.16. In the dialog box, browse to the project containing your desired macro library and browse into its 'lib' folder. The available libraries will be displayed in the Libraries white box. Select the library and close the library manager. The library will now be attached and visible in the Project Libraries dialog box (Figure 2.15). Now add that library in your current project libraries.

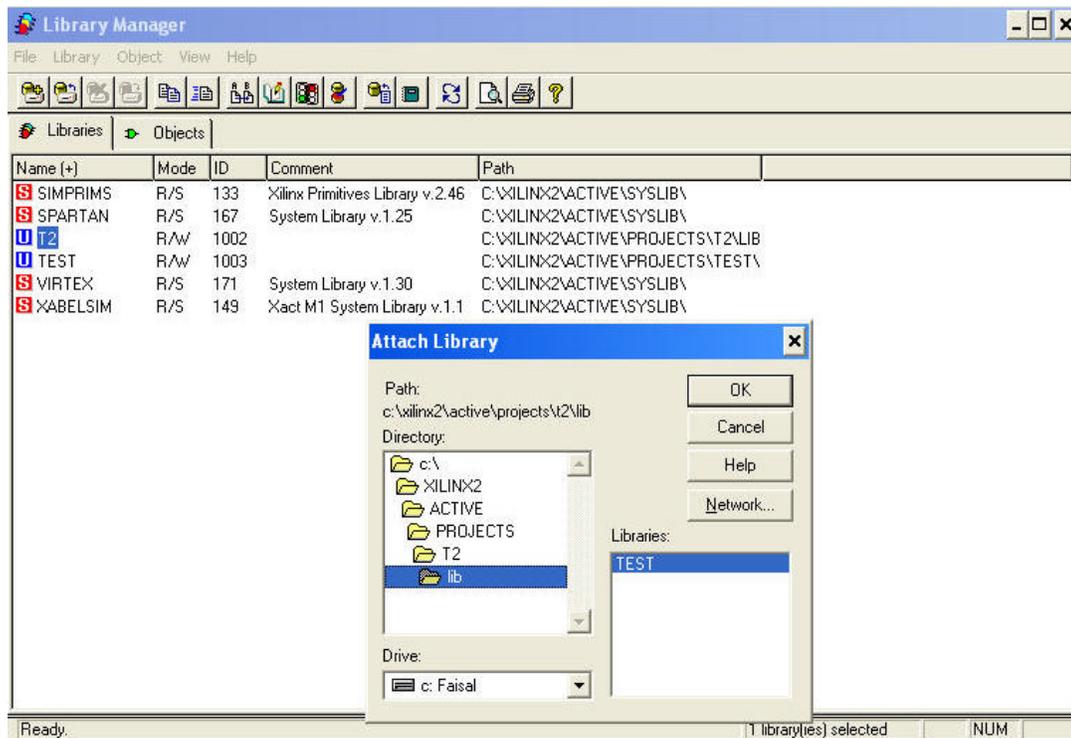


Figure 2.17: Attaching Library in Library Manager

The set of libraries that are attached with every project are stored in (Xilinx Installation Path\Active\Syslib). You can save attaching effort of your own developed library by moving your library to this location. To do this, right click on your library in Library Manager window and click on move (or copy). In the ensuing dialog box, select the default location for Xilinx libraries (Xilinx Installation Path\Active\Syslib).

Pre-lab

- Read the section on *Creating Macros*.
- Bring the schematic of the previous lab (verified and error free).

In-lab

The lab requires you to go through all the stages of the design process using Xilinx and Digilab tools. You will be designing and functionally testing your design in Xilinx CAD software. Then you will verify your design on the board as well.

- Start the Xilinx project manager, configure the device and name the project as lab4.
- Create schematic for the lab.
 - Start by importing the schematic of Lab3.
 - Create a macro for the circuit bounded by the shaded box shown in Figure 4.1 (Notice that this part is repeated three times in the circuit). Call this macro MACRO41.
 - Redo the circuit replacing the repeated parts with MACRO41.
 - Perform an integrity test. Check to see if there are any warnings and remove them.
 - Open a new schematic.
 - Add an input pad with its input buffers, and constraint it to SW1 (P28) as shown in Figure 4.2.
 - Repeat the last two steps for the other seven switches.
 - Do the same thing for the LED (LD1 to LD8) connecting output buffer to output pads and constraining them to the LDs as shown in Figure 4.3.
 - On a new schematic, connect the eight switch macros to the eight LED macros as shown in Figure 4.4.
- Verify functionality of the design.
 - Verify that the LED macros follow the state of the switches macros.
- Implement the design
 - Download the design on Digilab board
 - Verification

Bonus Staff:

- Analyze the circuit given in Figure 4.1 to find another region where you can use MACRO41.
- Build a two-bit adder using the circuit given in Figure 4.1 as a macro (FA) by connecting them as shown in Figure 4.5

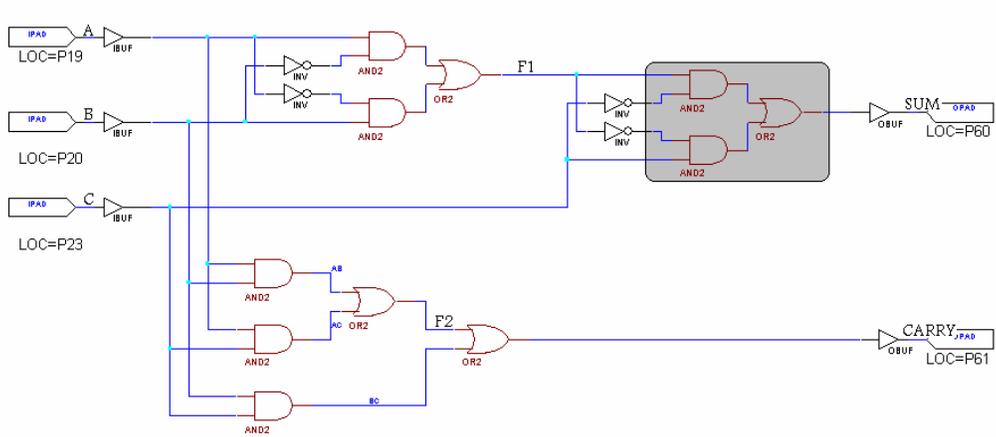


Figure 4.1

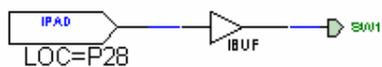


Figure 4.2



Figure 4.3

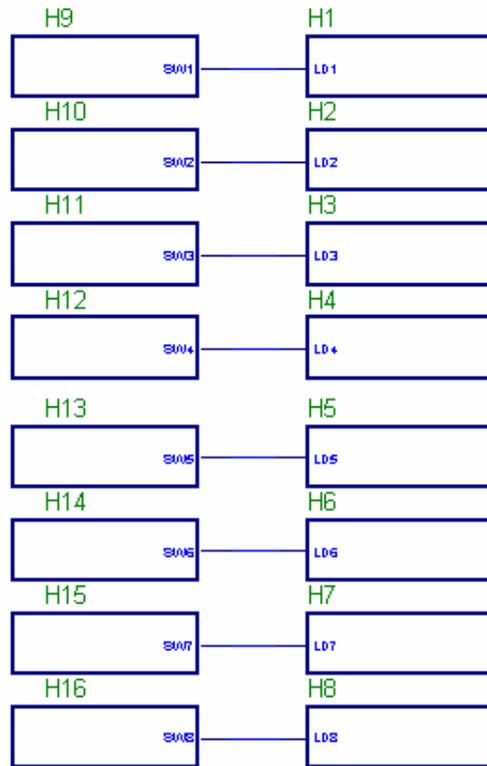


Figure 4.4

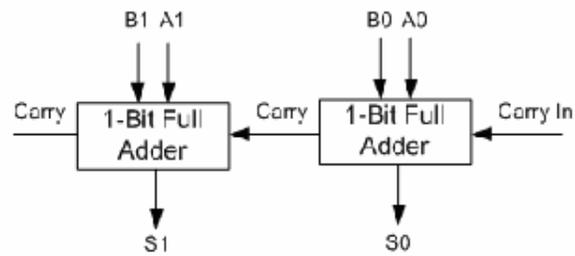


Figure 4.5