Main Design Project

Introduction

In order to gain some experience with using macros we will exploit some of the features of our boards to construct a counter that will count from 0 to 59 with the counts displayed in two of our four 7-segment displays. In addition we will instrument one of the sliding switches to enable counting with an light emitting diode (LED) indicating its state and we will add a push button to reset the counters momentarily to zero. The DI01 board is constructed in such a fashion that individual segments of the four displays are connected in parallel and are turned on when the particular connection is grounded. A given display may be selected for display by connecting its anode to a positive logic level. Read over the description of the DI01 module, in particular the description of the seven segment displays and how the individual segments combine to form numbers.

Macros

In your design you will need to use several macros one of which will include the 0 to 5 counter you constructed in last week's lab and another is a routine to translate the four bit output from the decimal counter into the pattern of 0's and 1's corresponding to the activated segments on a 7 segment display. This will be the file hex2led.vhd which you will construct. A third macro, provided for you, will divide the 50 MHz clock down to a frequency suitable for display and easy verification of proper opeation. On the basis of this macro you will write another (in vhdl) to generate a toggling frequency clock which will be an input to the macro multiplexing the two 4-bit output streams from your counter onto the 7-segment display.

Procedure

Open Project Navigator and start a new project choosing "schematic" as the highest level module. Verify that the Device Family is "Spartan2", the Device "xc2s30" and the Package "tq144".

The Counter

If you succeeded in building the 0-59 counter in last week's lab, add it to your current project (Project→Add Source) and create a Symbol out of it by double-clicking on "Create Schematic Symbol", while the source file is selected. Be sure to verify that this new symbol is available for use in future schematics. If you did not get that far last week, finish with the 0-59 counter before proceeding further.

The Hex to LED Converter

We will use VHDL to generate this macro.

1. In Project Navigator, select **Project→New Source**.
   The New Source dialog box opens.
2. Select the source type **VHDL Module**.
3. In the File Name field, type **hex2led**

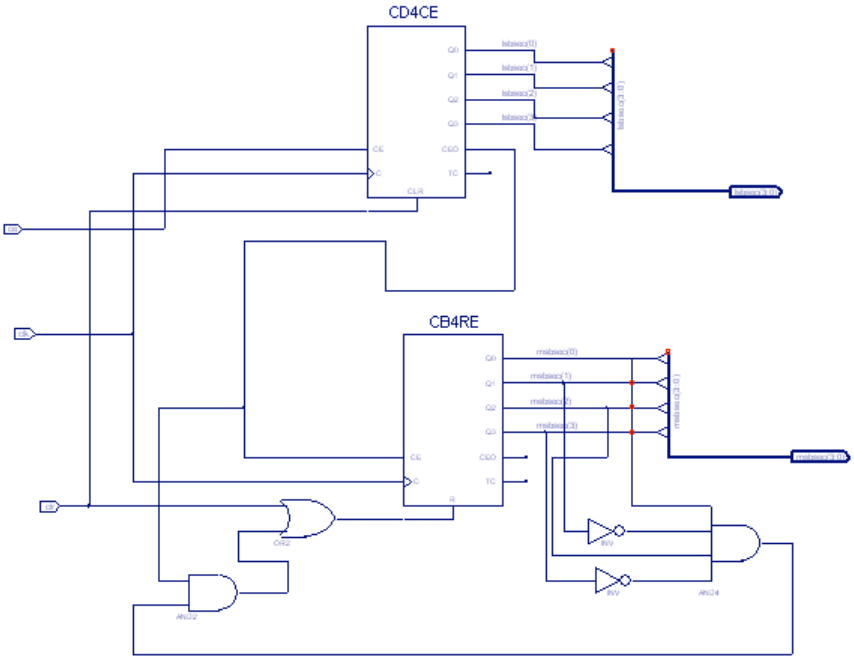

Figure 1: Counter Schematic Layout

4. Click **Next**.
   The hex2led component has a 4-bit input port named HEX and a 7-bit output port named LED. First enter the port named HEX as follows:
5. Click in the Port Name field and type **HEX**.
6. In the Direction field, set the direction to **in**.
7. In the MSB field, enter **3**, and in the LSB field, enter **0**.
8. Repeat the previous steps for the LED(6:0) output bus. Be sure that the direction is set to out.
9. Select **Next** to complete the Wizard session.
10. Select **Finish**. The "skeleton" HDL file opens in the ISE Text Editor.

In the HDL file, the ports are already declared and some of the basic file structure is already in place. Key words are displayed in blue, data types in red, comments in green and values in black.
Next we will use some synthesis templates to finish this design.

1. In Project Navigator, select **Edit→Language Templates**.
2. Locate the template called HEX2LED Converter for VHDL located under the Synthesis Constructs heading→Coding Examples→Misc.
3. To preview the HEX2LED Converter template, click the template in the hierarchy. The contents display in the right-hand pane.
4. Copy the contents of this template and add it to your hex2led.vhd file under the architecture begin statement.
5. Save the file by **File→Save**.

6. In Project Navigator, select hex2led.vhd in the Project window.

Double-click Check Syntax located in the Synthesize hierarchy in the Processes for Current Source window. This launches the ISE Text Editor.
If no errors are found, create a schematic symbol of this file by double-clicking "Create Schematic Symbol" and make sure it is available in the Symbols area on the schematic entry page.

The Frequency Dividers

Your TA will tell you where you can pick up the HDL file for dividing the clock frequency by 2^23. Examine the syntax of the file and then create another one to provide the frequency to toggle the displays. How high a frequency is reasonable for this function?

The Multiplexer

Design a macro to take as inputs two four-bit streams of bits and a toggle level and output one of the four-bit streams depending on the level of the toggle signal. You will need to use four 2 to 1 multiplexers from the Symbol repository. If you have time, create waveforms and run Modelsim to exercise this function. When you have finished your design without errors, make a symbol out of it for later use.

The Design

You now have all the pieces for the design, the counter, the multiplexer, the Hex to LED converter and the frequency reducers. Assemble the final design. Remember that your inputs will be the clock, the clear button and the counter enable button. Your outputs consist of the seven-segment counter and two toggle lines to choose a counter to display. If your design has no errors you can start tying up the outputs to the pins on the D2XL card via the user constraint file. Referring to the pin numbering given in the DIO1/D2XL pinout page, tie the clock to the 50MHz system clock, the seven data lines to the seven segment counter LED's, the CE line to a switch and the CLR line to a button. Generate the two lines to choose a display counter from the toggle clock, one from the clock itself and the other from its inverse. To indicate that the CE line is on, route its output to the LED opposite the switch. After you have assembled everything, your final design should look like that in Figure 2.

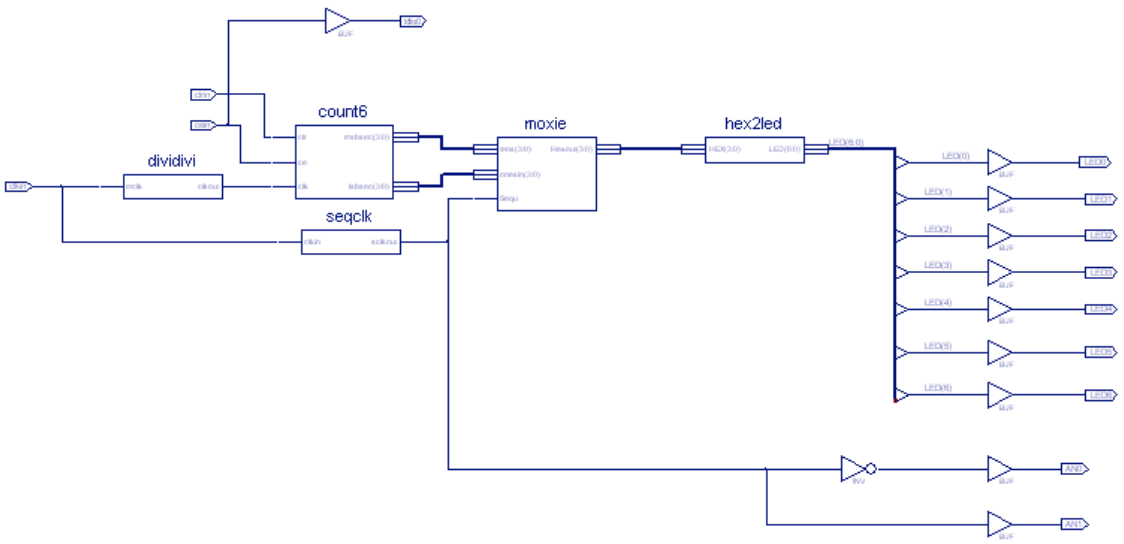

Figure 2: Schematic of Counter Design.

Exercise all the functions of your counter and demonstrate for your TA that the design works as advertised.