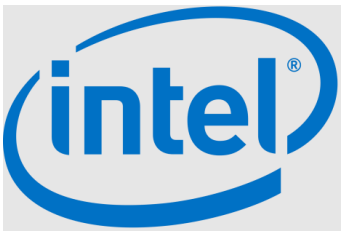

TEST YOUR KNOWLEDGE: CALCULATOR FOR HANDS-FREE REMOTE BOARD

To learn more, visit http://fpgauniversity.intel.com

Contents

BACKGROUND

Summary

This is a lab to test your knowledge in assembling a small FPGA calculator project on the DE10- Lite remote development kit. The circuit adds the values of two counters (0-98) that are enabled with push button (KEY) switches. The two addends and the sum are displayed on the seven segment displays. Since the sum can overflow to a 3rd hundreds place digit and there are only 6 seven segment displays, the hundreds position should be represented by lighting up all 10 LEDs simultaneously.

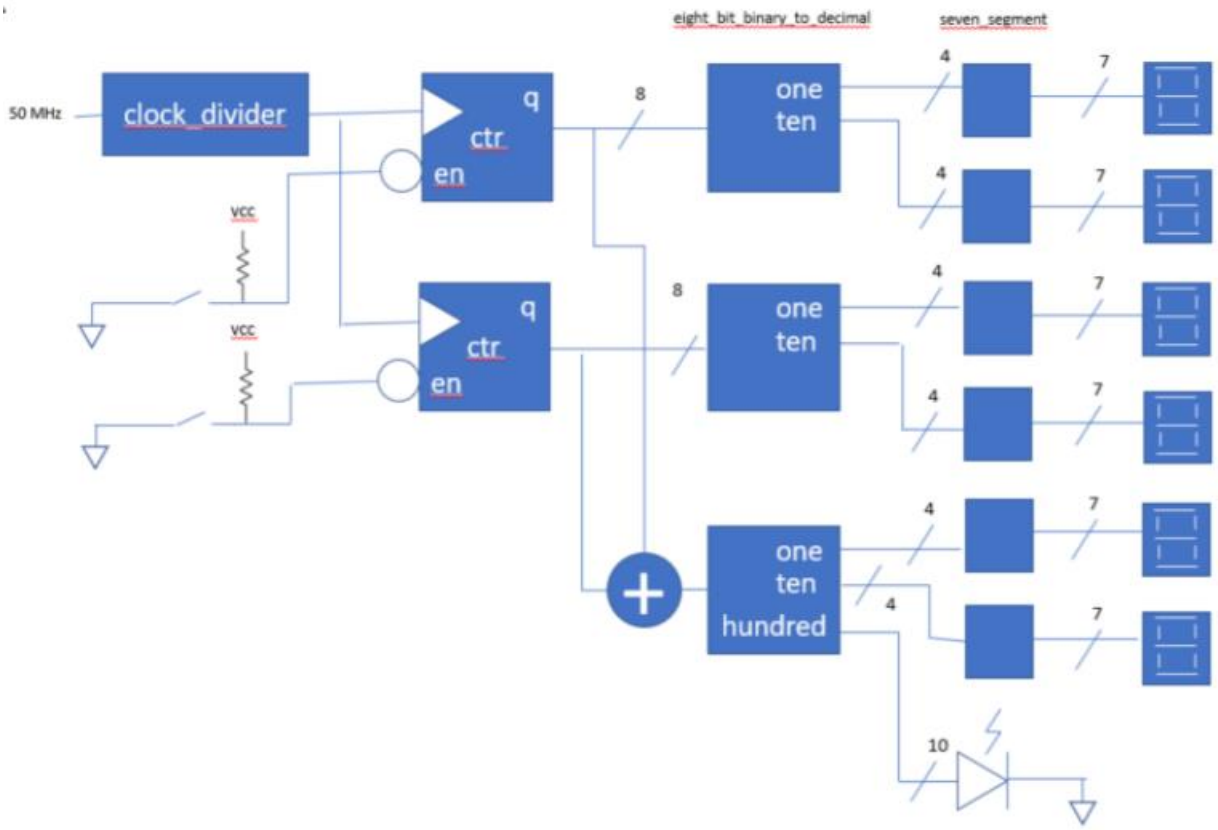

Figure 1: Calculator Block Diagram

LAB INSTRUCTION

Modules Provided

In order to complete this lab, you will need the following modules:

- Ctr 0-98: change to 0-99 (built from IP library)
- clock_divider: 50 MHz to 2 Hz
- eight_bit_binary_to_decimal: converts binary to decimal
- seven_segment: converts decimal to seven segment display

Project Setup

- To start you will need Calculator.qar
  To download the file click here: https://github.com/intel/FPGA-Devcloud/tree/master/main/HandsFree/Devkits/DE10-Lite/Calculator
- Open the .qar file by double clicking, or File → Open Project. Note the folder location where the information gets unarchived.
- A window will pop up on Quartus that says “Restore Archived Project,” click OK.
- To make sure all the needed files are in your project, go to Project → Add/Remove Files in Project… You should see the files shown in Figure 2 below

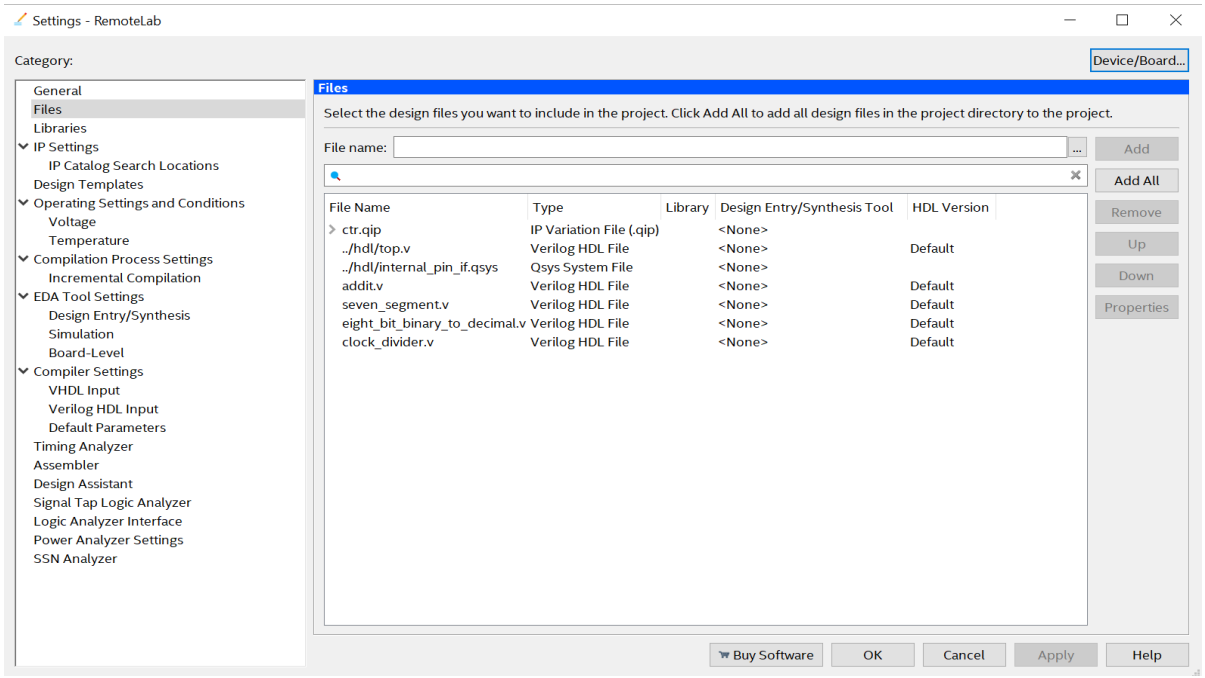

Figure 2: Project Files

Getting Started

The top level module wrapper has already been created for you, it's called top.v. You will add your design in addit.v. In Project Navigator, change Hierarchy to Files to open addit.v. Look at the Calculator Block Diagram and wire up the modules. It's very easy to mix up the wiring, so be careful and organized as you connect all the wires.

HINTS:

- ✓ Some hints and tips are already in the comments inside your code. Make sure to read them all.
- ✓ You do not need to edit or change anything from the given modules. Only addit.v and top.v
- ✓ In the first lab you learned that for the Remote Board you must instantiate your modules in the Top level. Don't forget to do it here.
    - Finish the following example for instantiation in your top.v file:

```
addit  i_addit(.clock(MAX10_CLK1_50),
               .push_button(KEY),
               //Add HEX displays and others
```

Compile your project and program. The remote board will open automatically after successfully compiling.

Once you got your counter compiled successfully and you have verified that your program is working correctly, close the Development Kit Viewer. Go back to your code and change ctr to count to 99.

Next, make the hundreds position light up 10 LEDs. Add LEDs to your top level, recompile and check your program.

Goodluck!

Bonus Section (Go for it!)

Make the two ctr modules count up or down, by controlling its direction with Switch 0 and Switch 1. Note that the counter is parameterized IP. To edit the counter: **Project Navigator** → **Change to IP Components**. Right click on LPM_COUNTER and change to

include up-down input port. You will need to add the up-down port to the top level and connect to appropriate SW[0] signals which are preassigned in the assignment editor.

DOCUMENT REVISION HISTORY

List the revision history for the application note.

Name	Date	Changes
Samantha Banda	8/10/2020	Initial Release of guide