

# Digital Counters

Publish Date: Dec 17, 2013

## Overview

Taking a hands-on approach to learning digital logic can be difficult without the need for students to learn complex hardware descriptive languages (eg. VHDL). Multisim Programmable Logic Diagram (PLD) along with support for leading Digilent teaching hardware allows students to put the fundamentals of digital theory into practice. The PLD schematic allows educators and students to create graphical logic diagrams like those found in textbooks and deploy these to Digilent educational boards. In this set of tutorials we demonstrate how digital logic theory can be taught using educational hardware to provide a hands-on approach to learning.

View the complete Teaching Digital Logic Fundamentals tutorial series (<http://www.ni.com/white-paper/14710/en>)

## Table of Contents

1. Introduction
2. Simulation Schematic
3. PLD Schematic
4. Simulating and Deploying Logic

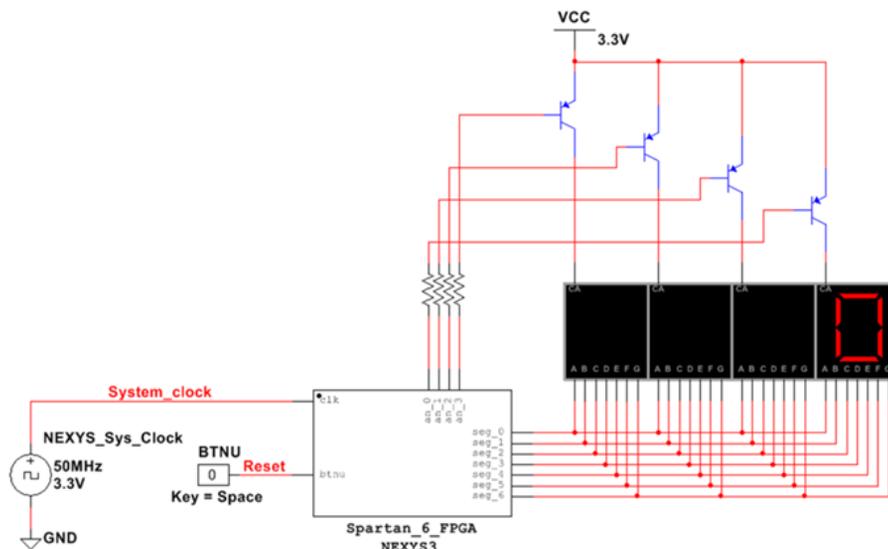
### 1. Introduction

When moving beyond simple logic diagrams, timing soon becomes a critical part of digital design. Gaining a comprehension of timing is difficult without using hardware because of the inability for software based simulations to meet the speed of those running in hardware.

Typically, using traditional approaches this would require the student to learn advance Hardware Descriptive Languages such as VHDL and Verilog. Using Multisim and the PLD schematic students can gain experience of using counters in hardware before the need to learn these more advanced descriptive languages. This tutorial is going to provide an example of how students can develop counters using Digilent boards and use these to control the onboard 7 segment displays.

Whilst this tutorial and the accompanying example were produced using the Digilent Nexys 3 the same could be achieved on other Digilent boards. More details on installing and targeting a range of Digilent boards from Multisim can be found here:

Getting Started with Digilent Boards in Multisim (<http://www.ni.com/white-paper/14871/en>).



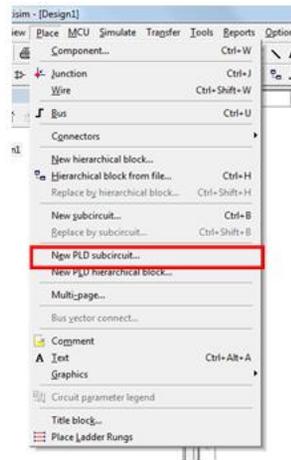
This tutorial will be broken down into three sections. Firstly we will discuss the top level schematic which is used for simulation purposes and contains the PLD subsheet. The next section will explain the PLD schematic and its various components. The final section will talk through running the logic both as a software simulation and on Digilent education boards.

### 2. Simulation Schematic

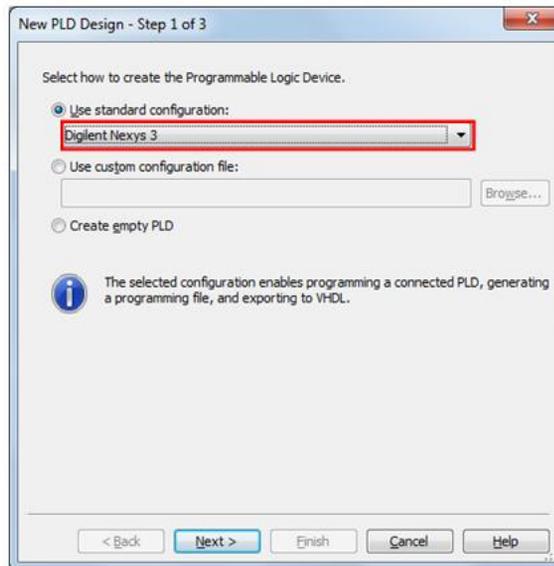
The top level schematic, as part of this project, allows for simulation. It places the PLD logic in place around the IO contained on the board. To achieve this we use a PLD sub-circuit. The PLD sub-circuit allows us to place the PLD code within a single component as if it was being run on the FPGA.

The steps below describe the process for creating a sub-circuit.

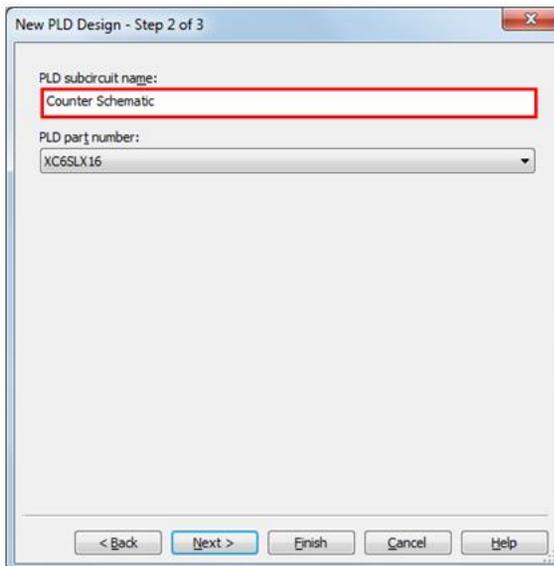
1. Open a new schematic.
2. Select **Place » New PLD sub-circuit**



3. Select the Digilent board you will be using from the drop down menu.



4. Provide a name for the PLD schematic.



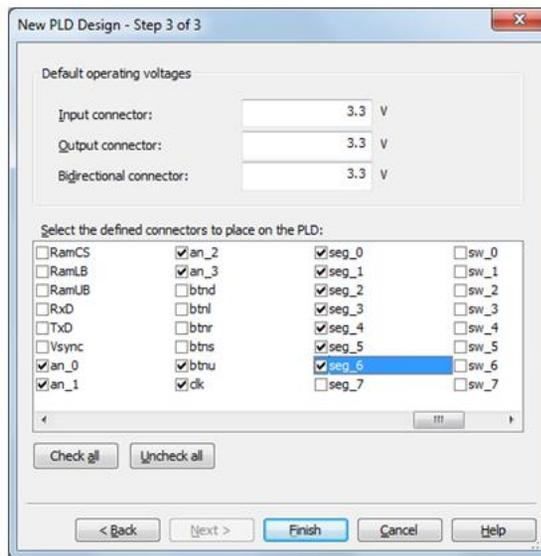
5. Select the connectors that you will be using on the PLC. For the counter in this tutorial we will be using:

**CLK – 50MHz clock on the board**

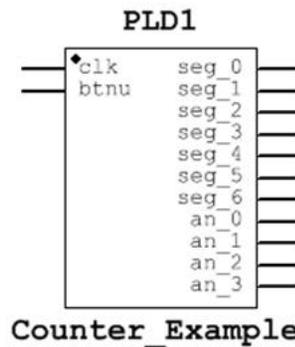
**BTNU – Push button located on the board to reset the counter**

**an\_0:3 – 4 Digital lines to control which of the four 7 segment displays to display**

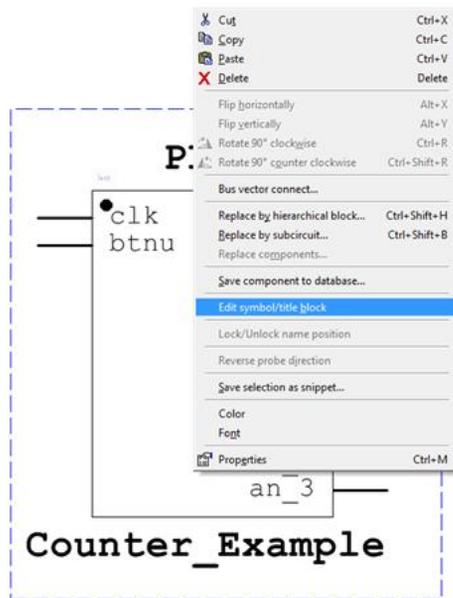
**seg\_0:6 – 7 Digital lines to control what each of the 7 segment displays displays**



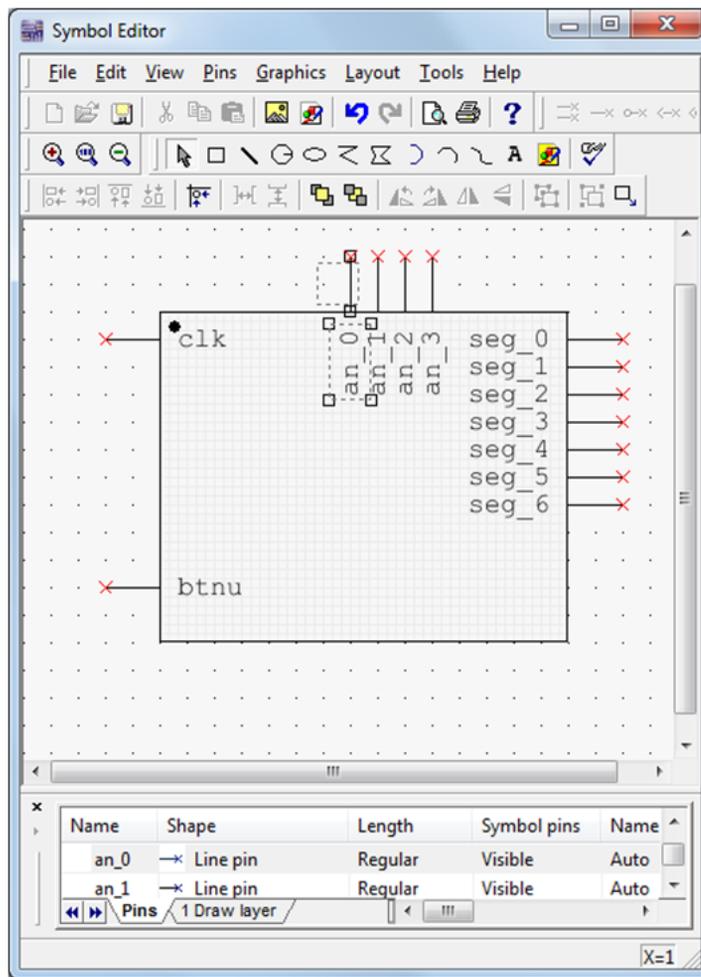
6. This will then generate the PLD block and you can place it on your schematic. This represents the FPGA on the Digilent board providing access to the various IO.



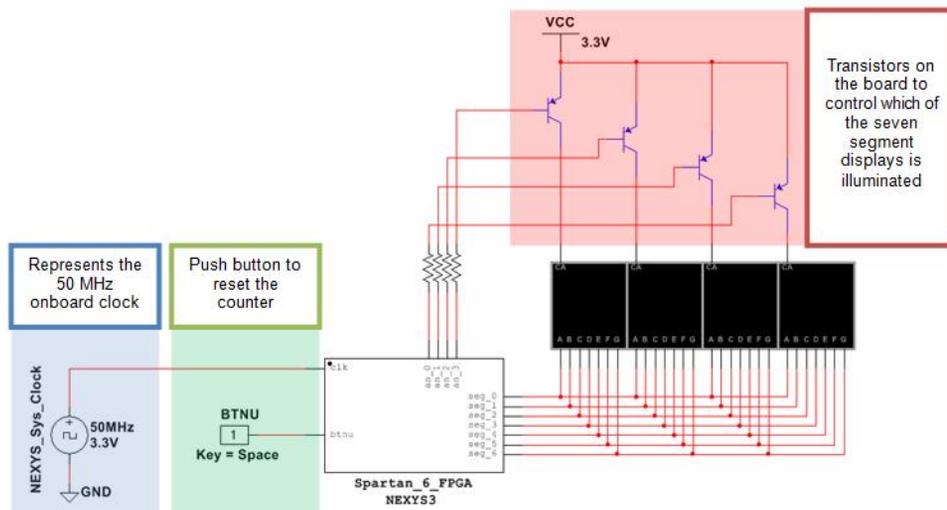
7. The symbol can be modified from the shortcut menu by selecting **Edit symbol/title block**.



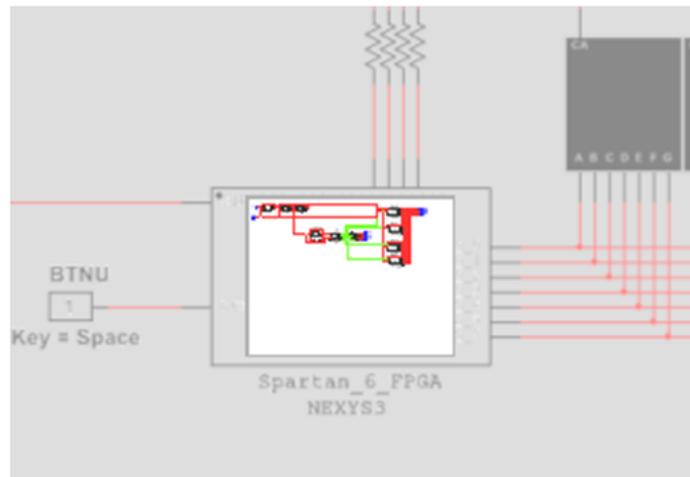
8. This opens the Symbol Editor that can be seen below where we can create a custom symbol and rearrange the terminals.



9. Save the symbol when complete and return to the schematic. From here we can create the components that feature on the Digilent board allowing us to simulate the schematic. The annotated figure below shows the required components for the simulation.



### 3. PLD Schematic

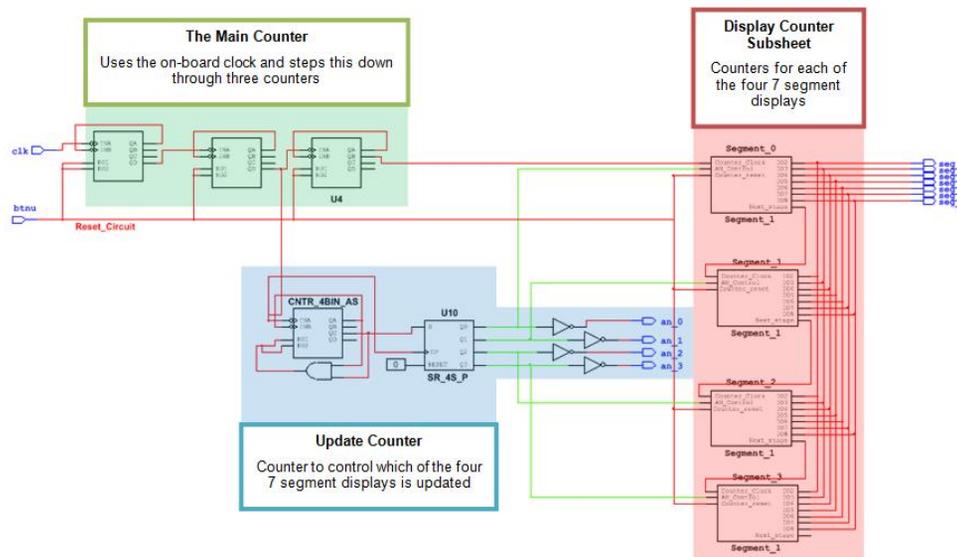


After completing the top level circuit we need to create the digital logic within the PLD schematic. In this example we are creating a simple up-counter and the required logic to control the four 7 segment displays. The PLD code needs to have two main functions:

- Control the illumination of the seven segment display
- Perform the up-counter operation.

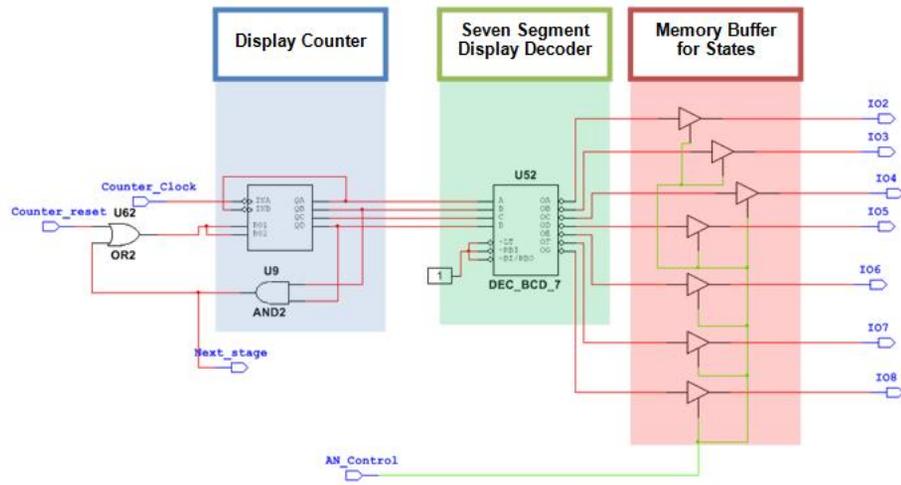
To create the PLD schematic double click the **PLD sub-circuit** and select **Open subsheet** from the properties window.

The figure below shows the three main sections of the PLD schematic. The main counter provides a clock signal to both the **update counter** and the **display counter**. The **update counter** controls which of the four displays needs updating. The **display counter** determines the value to be updated on that counter.



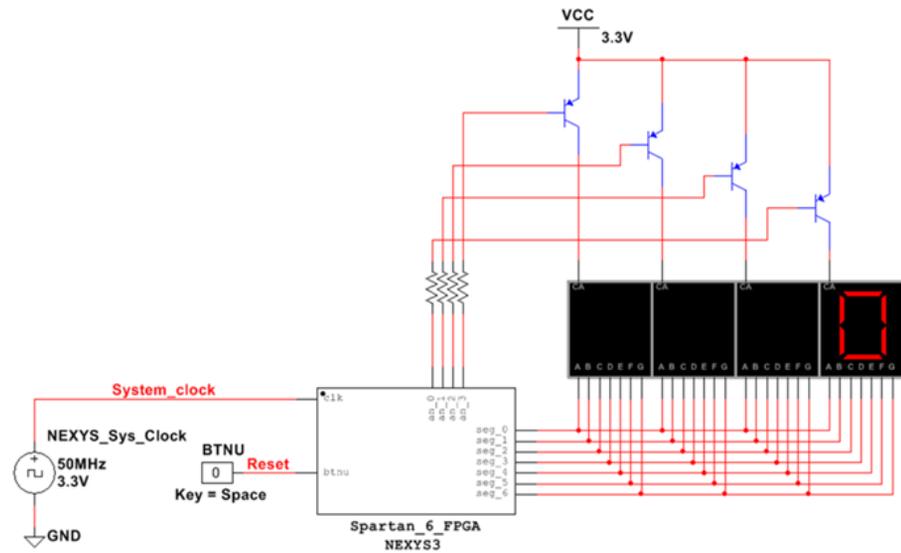
Each of the four **display counter** blocks contain a subsheet containing:

- Individual counter that calculates the value for that display before passing that value onto the next **display counter** block.
- A 7 segment-decoder to take the counter value and output the required states on to the 7 segment display.
- A buffer for each of the seven lines leading to the display.



#### 4. Simulating and Deploying Logic

The above PLD code can be simulated through running the top level circuit and viewing the display on the virtual 7 segment displays. During execution the seven segment displays are illuminated one at a time to display the count.

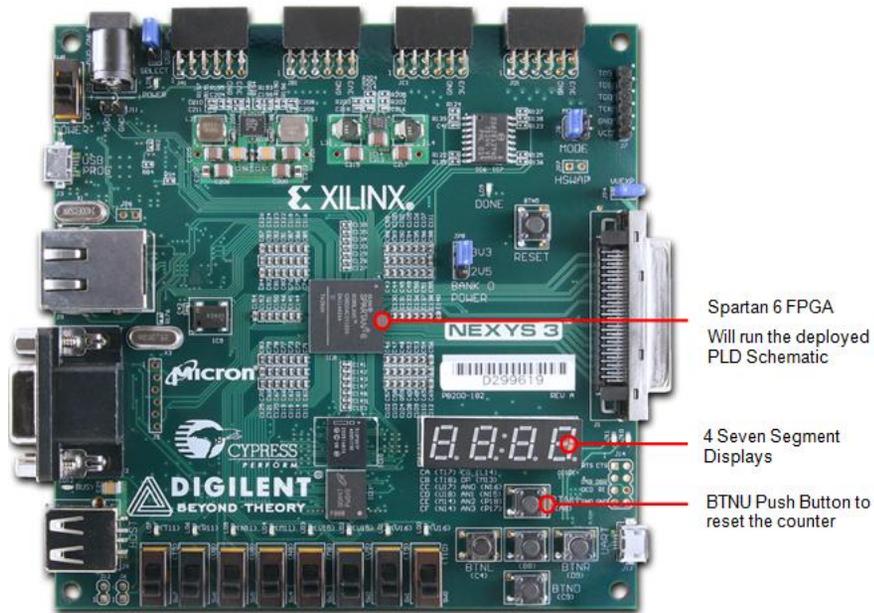


Within the simulated PLD code a shift register in the update counter section controls which of the seven segment displays is updated. This is then passed out of the PLD schematic and the transistors regulate which of the displays gets illuminated. The virtual BTNU button can be triggered to reset the counter.

The next stage of learning about the behavior of the circuit is to deploy the code to the Digilent board. This can be done by opening up the PLD subsheet and transferring to PLD as described in:

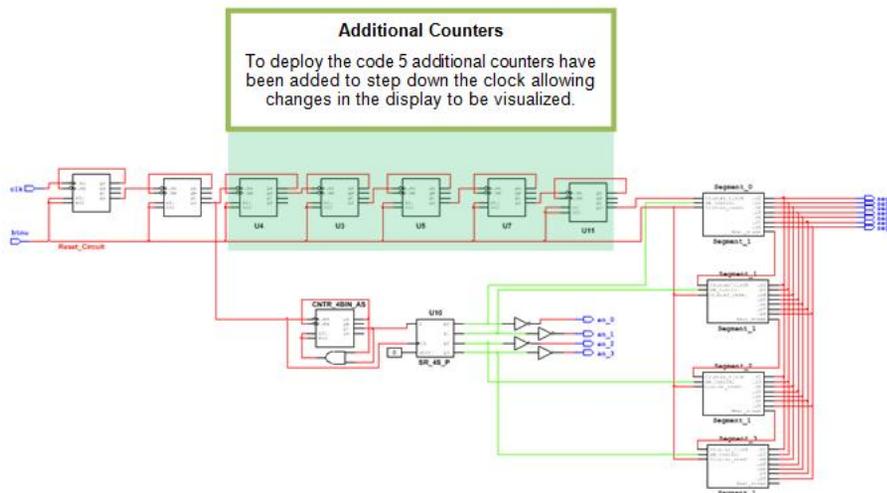
Getting Started with Digilent Boards in Multisim (<http://www.ni.com/white-paper/14871/en>)

The components featured on the top level schematic will now be replaced with the physical components on the board.

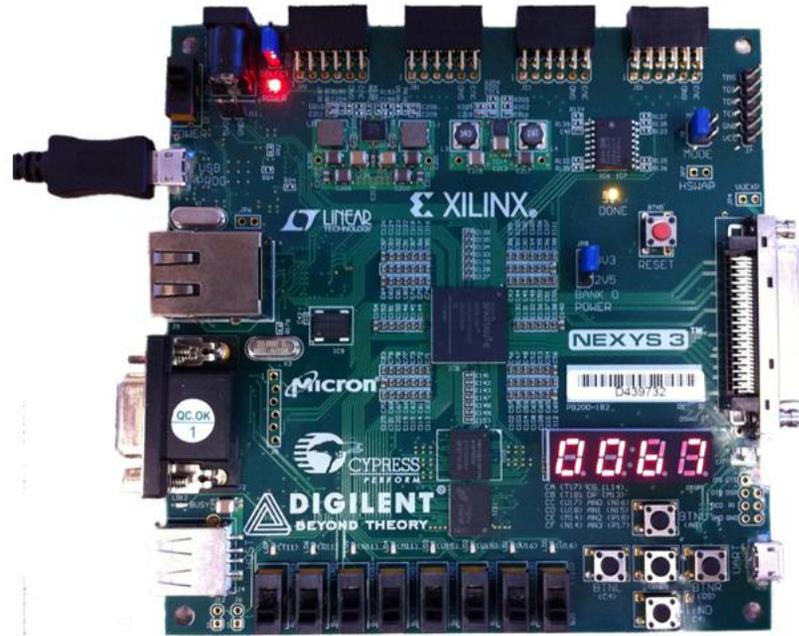


When deploying the PLD schematics featured above the student will notice that the seven segment LEDs appear illuminated constantly and it is not possible to view the count. This illustrates the FPGA on the Digilent board being able to run much faster than the software on the PC. This is the same with any software simulation of FPGAs and an important lesson which the students can learn first hand.

To overcome this we can step down the 50MHz clock further so that it is slow enough to allow us to view changes increments. This can be seen below with the addition of 5 counters 4-bit counters.



Once we deploy this logic we will see the 4 seven segment displays illuminated and performing the count operation.



All the files for this tutorial can be found in the attached file. This tutorial provides an example of how students can learn the fundamental theory behind digital logic counters and deploy these to hardware before the need to learn advanced Hardware Description Languages such as VHDL.

Download the tutorial files here ([http://vcm:27110/vcm-file-access/static/ftp/pub/gdc/epd/digital\\_counter\\_nexys3.zip](http://vcm:27110/vcm-file-access/static/ftp/pub/gdc/epd/digital_counter_nexys3.zip)).