ECE 4401 Lab 1

Fall 2014

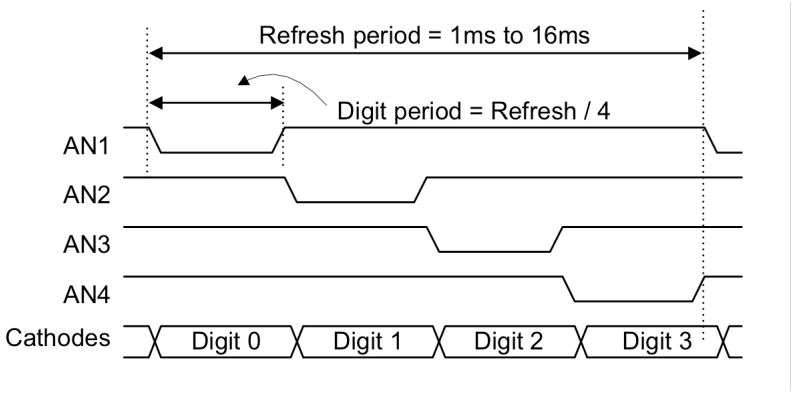
1 Introduction

This lab will introduce you to the Digilent/Xilinx Nexys 2 Board. You should have a copy of the *Nexys2 Reference Manual*. A quick overview of the Xilinx ISE design software is given in this lab document, but it is assumed that you have some experience with the tools from ECE3401.

The basic goal of this lab is to design logic to control the 7-segment LED display. The project specifications are as follows:

Use BTN3 to reset the board. On reset, the 7-segment LED display should read **0000**. Then, the LED display should start incrementing its values every second depending on the value of the switches. If SW0 is off, the LED display will increment in decimal; if SW0 is on, the LED display will increment in hexadecimal. Note that toggling SW0 should toggle the LED display between the same number in hex and decimal representations.

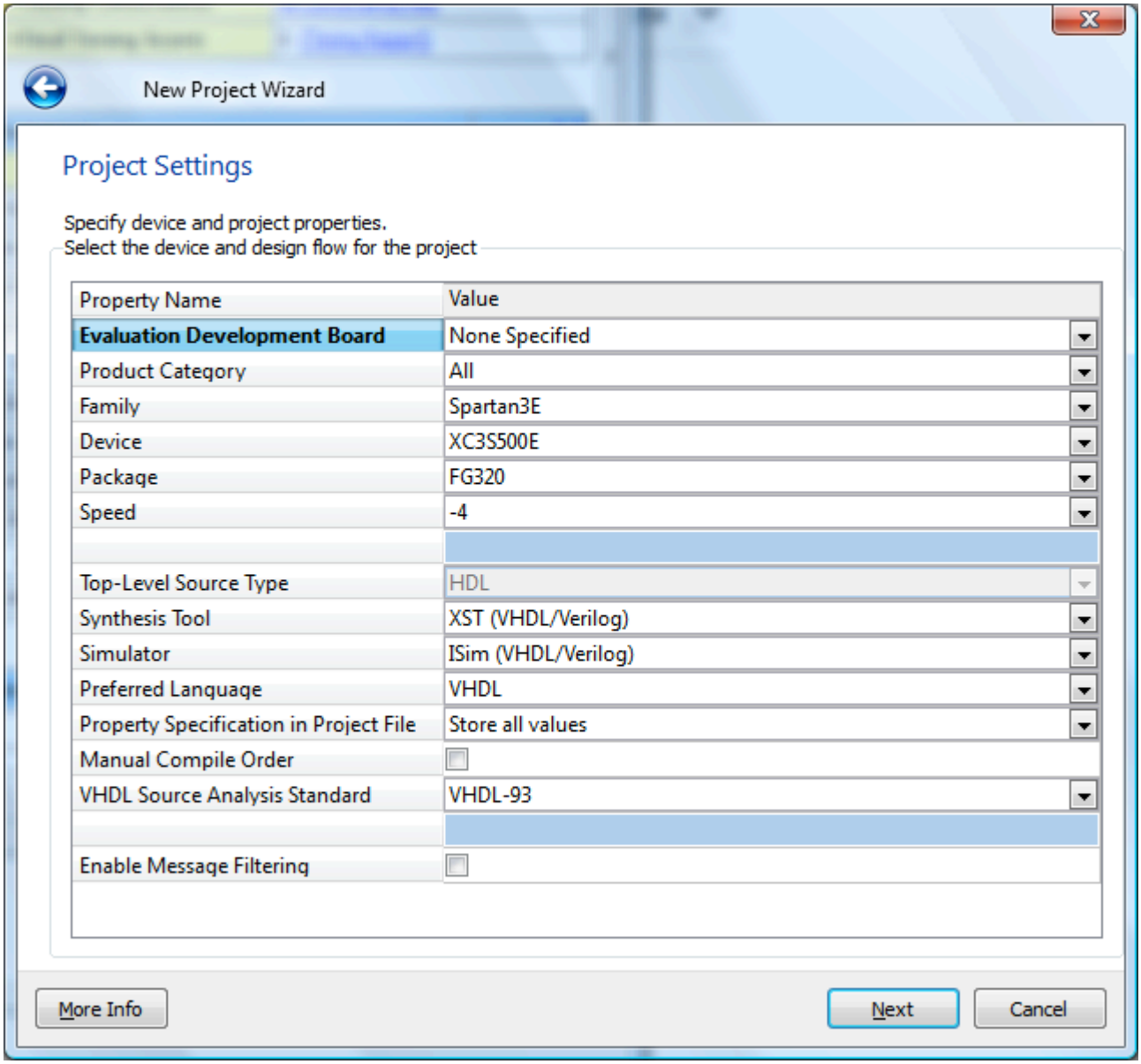
The LED display operation is given on page 5 and 6 of the *Reference Manual.* The LED display is controlled in a time-multiplexed fashion by activating one digit at a time using the `AN` digit control lines. The figure below shows the clock waveform for driving the LED display. In order for the LED display to respond properly to the control signals, each `AN` pulse should be approximately one millisecond long. The operating frequency of the board clock is 50MHz.



2 Setting up the ISE project

Open up the Xilinx ISE Project Navigator. The icon should be on the desktop or available under the ***Xilinx Design Tools->ISE Design Suite->ISE Design Tools*** menu in the Programs listing in the Start menu. The first step is create a new project by selecting ***File->New Project…*** You can name the project anything you wish (lab1 is a good choice) and choose HDL as the top-level source type. The project location should be on your personal directory mounted from the engineering file server or your

own flash drive. If you save the project on the local disk of the computer, it may disappear the next time you log in. Click *Next* and you are given an opportunity to select the device and design flow for the project. Fill in the values as shown on the next page.



You can click *Next* and then finally click *Finish* to complete creating the new project.

Once you have created the project, you can now create your toplevel VHDL file. Right click on the project name in the left top panel and choose *New Source*…. In the window that pops up, choose *VHDL Module* as the source type. The file name can be anything, but for consistency give it the same as the name of your project file. The location should be the same as your project. Click *Next* and then on the next screen, enter the port names if you wish at this time and click *Next* again. You can click *Finish* to create a template VHDL file. For this lab, you will need inputs corresponding to the system clock, the reset button, and the SW0 switch. For outputs, you will need the segment display signal outputs.

The next step is to add a user constraints file that describes the connections of the FPGA. Download the `Nexys2_500General.ucf` file from HuskyCT and put it into your project directory. Then, from the Sources panel, right-click and select "Add Copy of Source". Choose the `Nexys2_500General.ucf` file and click *Open.* You should not change the pinout definitions given in the `Nexys2_500General.ucf` file since they correspond to the actual wiring on the Spartan-3 board. Change the port names in either your toplevel vhd file or in the `Nexys2_500General.ucf` file so that they match. Note, that vector arrays are explicitly listed in the `Nexys2_500General.ucf` file. For example, the lines

```
NET "Led<0>"  LOC = "J14";
NET "Led<1>"  LOC = "J15";
NET "Led<1>"  LOC = "J15";
NET "Led<2>"  LOC = "K15";
NET "Led<3>"  LOC = "K14";
NET "Led<4>"  LOC = "E17";
NET "Led<5>"  LOC = "P15";
NET "Led<6>"  LOC = "F4";
NET "Led<7>"  LOC = "R4";
```

correspond to the `Led` vector array that you can use in your toplevel VHDL.

You can now write the VHDL to implement the LED display control described above. Remember to think about modularity. There will be many labs that will require LED display and clock manipulation. Placing that functionality into separate modules will help you in future labs. As you did in ECE3401, you can use Isim or Modelsim and a testbench VHDL file to simulate the circuit.

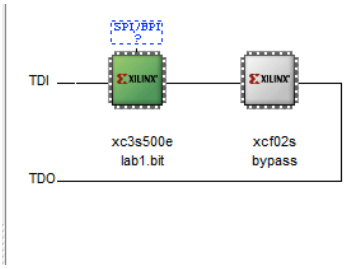
3 Programming the Nexys2 Board

Once you have finished writing your VHDL and simulating it using VHDL, you can download your design to the actual FPGA board. The first step is to generate the programming file. Click on the top-level file in the source panel and the processes should appear in the panel below.

First we need to set some options correctly, right click on *Implement Design* in the Process View and selecting *Process Properties*…. In the window that pops up select the checkbox for *Allow Unmatched LOC Constraints* and click OK. Then, right click on *Generate Programming File* and select *Process Properties* …. In the window that pops up, click on Startup Options and change the Startup clock to JTAG clock.

Now, you can go ahead and double-click on *Generate Programming File* process. At the end of the *Generate Programming File* process, you will have a .bit file that you can download to the board. Make sure the USB cable is connected from the computer to the USB connecter on the Nexsys2 board.

Once you have the cable connected, click *Manage Configuration Project* which will open up the iMPACT program. If it has connected to the board properly, you will see the following display:



Righ click on the xc3s500e chip and select *Assign new configuration file* and choose the lab1.bit file. Then right click again, and choose Program to download the bitstream file to the FPGA. If everything works right, you should see the numbers on your seven-segment LCD display.