

Lab 1 - A Two-Bit Counter and LED Driver

Introduction

The purpose of this lab is to design, simulate and implement a two-bit counter that drives a seven-segment LED display. You will design the device using VHDL and implement it on a programmable logic device called an FPGA (field-programmable gate array).

The Lab Equipment

The lab equipment consists of a single-board computer using an Intel 386EX processor, a PLD (programmable logic device) evaluation board with an Altera FLEX10K-series “20,000-gate” FPGA, a wireless prototyping area, a PC-based logic analyzer, a power supply and a PC.

In this lab we will only use the FPGA board and the PC. The other components will be described in later labs.

The FPGA Board

The FPGA on the PLD board has 240 pins which can be configured as inputs or outputs. Two of these pins are connected to two pushbutton switches on the board. These will act as the two inputs for your design. There are also two seven-segment LEDs connected to the FPGA. One of these will be the output for your design. The pin numbers are as follows:

Function	FPGA Pin
pushbutton 1	28
pushbutton 2	29
LED segment a	6
LED segment b	7
LED segment c	8
LED segment d	9
LED segment e	11
LED segment f	12
LED segment g	13

Inputs and Outputs

The circuit has two inputs: a reset signal and a clock. The circuit has seven outputs, one for each of the LED segments. All signals are active-low. For example, an LED segment will be on when its corresponding output is set low ('0' in VHDL).

Required Behaviour

The output should change only on the falling edge of the clock. If the reset input is asserted (low) the LED should display the digit zero. Otherwise the LED should show the next higher value. If the output is already at 3, it should remain at that value (i.e. not “wrap around”).

Compiling and Simulating

The Max+PlusII software can be run on a PC under Microsoft Windows (Select the menu item Start|Programs|Max+PlusII) or on Sun workstations running SunOS 5 (Solaris) (run /nfs/wilton/export/altera/maxplus2/bin/max2win).

Entering your VHDL Code

Select the menu item File|New. Click on Text Editor File. Click on OK. A text editor window will be displayed.

Select the menu item File|SaveAs. Enter the name your vhd file. The VHDL file name must be the same as the name of the top-level VHDL entity in the file and have an extension of .vhd. Click on OK

Enter your your VHDL code in the text editor window. Note that the editor recognizes VHDL syntax and colours your text accordingly. Select the menu item File|Save when you are done.

Select Project Name

You must assign your project a name. This must be the same name as the top-level file name (omitting the .vhd extension). Select the menu item `File|ProjectName`. Select the desired file name. Click on `OK`.

Assign Device

Select the menu item `Assign|Device`. A dialog box will be displayed. Select the `FLEX10K` family and the `EPF10K20RC240-4` device. Click on `OK`.

Assign Pins

Select the menu item `Assign|Pin, Location, Chip`. A dialog box will be displayed.

Click on `Search`. A dialog box will be displayed. Click on `List`. Select the name of the signal you want to assign to a pin number. Click on `OK`. Select the pin number. Select whether it should be an `Input` or `Output` pin. Click on `Add`. Repeat these steps to add the rest of the pins.

Click on `OK`.

Compile

Select the menu item `Max+PlusII|Compiler`. A dialog box will be displayed.

Select the menu item `Processing|Timing SNF Extraction` if necessary to make sure this option is enabled.

Click on `Start` to compile your VHDL code. If any errors are displayed in dialog box, fix the errors and Click on `Start` again. Repeat until there are no errors. Use the on-line help if necessary.

Create Test Waveform File

Select the menu item `Max+PlusII|Waveform Editor`. A waveform editor window will be displayed.

Select the menu item `File|End Time`. A dialog box will be displayed. Enter desired range of time (e.g. 5us to give 50 steps of 100 ns each).

Select Signals

Press the right button in the blank signal name area to bring up a menu. Select the menu item `Insert Node`. Click on `List`. Select the signal you want to add. Click on `OK`. Repeat for all other signals you want to add (both inputs and outputs). It may be more convenient to use the (un-subscripted) group name for bus (vector) signals (e.g. use `led` instead of `led0, led1, ...`).

Set Default Values

Press the right button on a signal name to bring up a menu. Select the menu item `Overwrite`. Select a default signal value from the menu options.

Edit Test Waveforms

Select part of a waveform by dragging the mouse while pressing left button. Press the right button to bring up a menu. Select the menu item `Overwrite`. Select the new signal value from menu. Repeat for all waveforms. Select the menu item `View|Zoom In` or `Zoom Out` to adjust the extent of the display.

Select the menu item `File|Save As`. Select the project file name with an extension of `.scf`. Click on `OK`.

Simulate

Select the menu item `Max+PlusII|Simulator`. Click on `Start`. The simulation will run and the results will be displayed in the waveform editor.

Save and Print your Work

Save the files *projectname.acf* (device and pin assignments), *projectname.vhd* (VHDL code), and *projectname.scf* (test waveforms) to a floppy disk to bring with you to the lab. Print out the VHDL code and the simulated waveforms.

Program the Device

This can only be done in the lab.

Select the menu item `Max+PlusII|Program`. A dialog box will be displayed. Select the menu item `JTAG|Multi-Device Chain` if necessary to make sure this option is enabled.

Select the menu item JTAG|JTAG Chain Setup. A dialog box will be displayed. Click on Select Programming file. A dialog box will be displayed. Select *projectname.sof*. Click on OK. Click on Add. Click on OK.

If the Hardware Setup dialog box comes up, select the ByteBlaster and LPT1 options.

Click on Configure

You can now test your circuit by pushing the buttons on the FPGA board and looking at the LED.

Pre-Lab Assignment

You must design the circuit and verify its operation by simulating it before starting the lab. The TA will ask to see your VHDL code and simulation waveforms at the start of the lab.

In-Lab Demonstration

When your device is working properly, ask the TA to check your work. He will make sure your device works as required and ask you one or two questions to verify your understanding of the material.

Report

Submit a short report with a written description of your circuit together with a listing of the VHDL code and a printout of the simulation waveforms that demonstrate correct operation of your device.