Lab 1 - Counter and LED Driver

Introduction

The purpose of this lab is to design, simulate and implement a 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 ("SBC") 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.

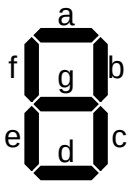
In this lab we will only use the FPGA board and the PC. The other components will be described in later labs.

The FPGA Board

The FPGA on the PLD board has 240 pins which can be configured as inputs or outputs. Four of these pins are connected to pushbutton switches (Pushbuttons A, B, C, D). Two of these will act as the two inputs for your design. There are also two seven-segment LEDs (FLEX_DIGIT1 and FLEX_DIGIT2) connected to the FPGA. The first of these will be the output for your design. The pin numbers are as follows:

Function	FPGA Pin
Pushbutton A	186
Pushbutton B	187
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

The LED segments are labelled as follows:



Inputs and Outputs

The circuit has two inputs: a preset signal and a clock. Use Pushbutton A (white) for the clock and Pushbutton B (black) as the preset input. 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

If the preset input is asserted (low) the LED should display the digit four (4). Otherwise the LED should show the next lower value. If the output is at 0, it should "wrap around" to four.

Your design must also be synchronous. The output should only change on the rising edge of the clock.

Compiling and Simulating

Start `Max+PlusII`

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) (see the instructions on the course Web page).

Enter 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 of your VHDL 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 the 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|Project|Name. 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.

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.

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.

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). Click on OK.

Select Signals

Press the *right* mouse button in the blank signal name area to bring up a menu. Select the menu item Insert Node. A dialog box will be displayed. 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 mouse button on a signal name to bring up a menu. Select the menu item Overwrite. Select a default signal value from the menu options. Use the Clock option to generate a clock. Repeat for all other signals.

Edit Test Waveforms

Select part of a waveform by dragging[1] the cursor in the waveform area to select a range. Press the right button in the selected area 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. Cor-

[1] Move the mouse while holding down the left button.

rect errors and recompile until you get the desired results.

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 3.5" floppy disk to bring with you to the lab. Print out the VHDL code and the simulated waveforms.

Program the Device

This part of the lab must be done in the lab.

Use Windows Explorer (or a DOS shell) to create a subdirectory for yourself in the max2work directory. Copy your files from the floppy to this directory. Note that the ACF file is an ASCII file that can be edited to change the project name and pin assignments. However, *do not edit the ACF file while* Max+PlusII *is running or the file will be corrupted*! If you created the .acf file under UNIX you will have to load and save it using the DOS Edit program to convert the file to DOS format.

Start Max+PlusII. Select the menu item File|Open. Select your VHDL file name. Click on OK. Select the menu item File|Project|Set Project to Current File. Compile your code as before.

Select the menu item Max+PlusII|Programmer. 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|Multi-Device JTAG Chain Setup. A dialog box will be displayed. Click on Delete All if necessary to remove any files already listed. 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.

Make sure the power supply is turned on. Click on Configure. It will take a few seconds for the FPGA to be configured ("programmed").

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

Make any changes required for your circuit to work properly. You can print a final copy of your VHDL code and simulation results on the lab printer.

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. It should include:

- a title page,
- a brief written description of the purpose and behaviour of your circuit (what it does, not how it does it),
- a listing of the VHDL code with comments describing each of the port signals and the purpose of each part of the code),
- a block diagram corresponding to your VHDL code,
- a printout of the simulation waveforms that demonstrate correct operation of your device.