

Laboratory I
CS-345, Fall 2004
Christopher Vickery

Contents	
Introduction.....	1
Method	1
Results	2
Discussion.....	2

Introduction

The purpose of this assignment was to become familiar with the FPGA Laboratory environment and the basic procedures involved in managing computer accounts and working with the Celoxica RC200E circuit boards, the DK Integrated Development Environment, and the Handel-C hardware implementation language.

Activities included becoming familiar with using Windows XP accounts with a remote server maintaining user preferences and document directories, using Celoxica-supplied example projects to implement designs both for simulation and for downloading to the RC200E boards, and modifying one example project as a way of introducing some of the features of the Handel-C language.

Method

The first step was to log into our Windows XP account in the CS Department's TREE domain, to provide a new password for the account, and to use the Firefox web browser to access the handout for the laboratory.

Celoxica provides a workspace with several example projects that use the PAL (Platform Abstraction Layer) in one of the installation directories of the Celoxica software, ...\\PDK\\Examples\\PAL. Because student accounts do not have the access rights that would allow modifying this workspace, we made a copy of it a subdirectory of the "My Documents" folder where student accounts are allowed to make modifications. We created a "My Projects" folder to hold all the projects for the semester, and copied the PAL workspace there, creating a directory named ...\\My Documents\\My Projects\\PAL.

We noted how long it took to log off and back in again with the PAL directory under My Projects, with it moved to the remote file space on the network drive H:, hosted on maple. We then moved the PAL subdirectory back to the local file system for the remainder of the laboratory session.

We ran the DK IDE by clicking on the PAL/Examples.hw file, which opened the Examples Workspace containing 19 projects provided by Celoxica. After setting some editor settings so it would be easier to write code that adheres to the Coding Guidelines for the course, we activated one of the example projects, SevenSeg, by selecting it from

the drop down list of the IDE. We selected the Sim build configuration, and modified the build settings to change some relative pathnames that referred to files in the installation directories so they could be accessed from the new location of the workspace using absolute pathnames.

We built and ran the sample project as a simulation, and again using the RC200E build configuration, which produced a .bit file that we downloaded to an RC200E using the FTU2 utility that is part of the Celoxica software installation.

We then modified the sample project so that two seven segment displays showed the value of an 8-bit counter in hexadecimal. Pressing one button on the side of the RC200E caused the counter to increment every half-second, and pressing the other button caused the counter to decrement every half-second. Pressing both buttons simultaneously, or neither button, caused the counter value to remain unchanged.

Results

Logging into the Windows XP account went smoothly enough, but there were problems with the setup of the Firefox web browser. It was necessary to run the browser with the “-profilemanager” option and to create a Default User profile, after which the browser worked properly.

Although we did not measure the difference precisely, it was clear that it took longer to log off and back on when there were several files and folders that had to be copied between the local computer and the server (when PAL was under My Documents) compared to when the same files were stored on the server and didn’t need to be copied back and forth every time we logged off and on.

When we simulated the SevenSeg project we observed that two windows were displayed. In addition to a “command prompt window” used for the simulation programs *stdin* and *stdout* there was a “PAL Console” window that had 8 seven-segment displays, 8 pushbuttons, 8 check-boxes, and some unused space that remained blank. Since the RC200E has only two buttons and two seven-segment displays in addition to several components we haven’t investigated yet, we could see that the PAL simulator can handle circuit boards more complex than the one we are using and probably can handle parts of the RC200E that we aren’t using yet.

We modified the sample code to work with switches to control the count direction, trying to mimic the way seven segment displays were managed in the sample code. Ultimately, the project worked correctly, both for simulation and when downloaded to the RC200E.

Discussion

Working with Windows XP in a networked environment requires care when deciding where to store files and folders. Keeping them on the local system speeds up interactive work, but involves the overhead of copying them to and from the server every time we logged on or off. Keeping them on the server’s disk (H:) saves the overhead when logging on and off, but requires network overhead every time a file is accessed while developing a project. A good strategy would seem to be to keep just the workspace containing the projects under active development under My Documents\My Projects, and to keep completed projects and any other less-frequently used material on the H: drive.

It is clear the Handel-C language can be used to program a field programmable gate array to do “interesting” things and to generate a Windows program to verify the program’s correctness without going through the overhead of generating a .bit file. But many mysteries remain about how the language works and how to use it.