Laboratory I

September 10, 2003

Objective

The objective of this session is to become familiar with the Celoxica DK development platform, using sample Handle-C projects provided with the kit. When you finish this lab, you should be able to configure a project so that you can either simulate it using the simulator provided with the Platform Development Kit (PDK) or download it to an RC200E for testing.

Lab Activities

1. Activate your computer account.
2. Configure your DK settings.
3. Copy a sample workspace to your account.
4. Configure your copy of the workspace.
5. Simulate one or more of the workspace examples.
6. Synthesize one or more of the workspace examples.
7. Test one or more of the workspace examples on an RC200.
8. Modify one or more of the workspace examples.
9. Submit a report of your lab activities.

Activate your computer account

You will work with a lab partner this evening, but both of you have separate accounts on the laboratory computers. For this lab, work through all the lab activities using one of your accounts, and then do the first three activities for the other account.

Your account name is the first five letters of your last name, and your password is *chip03*. You have a network account in the *TREE* domain. Log into your account, and change the password to something secure.

Customize your Windows XP settings to suit your preferences. Be sure you can log out and in again okay.

Configure your DK settings

Start DK. There should be a shortcut to it on your desktop. If not, use Start → Programs → Celoxica → DK Design Suite → DK. If you like, you can right-click the DK shortcut and make a copy on your desktop. Also, you can right-click the shortcut and assign a shortcut key to it.

If you have trouble with the licensing when you start DK, let me know. The license server is oak.cs.qc.edu if you run the software from off campus.

From within DK, select Tools → Options, and click on the Directories tab. You need to add the following path to the directories for Include Files:

C:\Program Files\Celoxica\PDK\Hardware\Include. You should use the Browse button to select this path to be sure you don't make a typing error.

Now switch from Include Files to Library Modules in the Directories tab, and be sure the one pathname there is *C:\Program Files\Celoxica\PDK\Hardware\Lib.*

Copy a sample workspace to your account

You have a network-wide home directory on the H: drive. If you click on "My Computer" you will a network drive called "courses on maple (H:), and under that you will see a directory with your account name, which will be your home directory for the course. Create a directory under your home directory for your projects for this course called "My Projects." No matter what computer you use, you will be able to access your projects from this location. Do *not* use the "My Documents" directory, which belongs to you, but is located on the C: drive of each different computer. Anything you put there will not only be inaccessible from other computers, but also will get lost whenever we restore the system.

Copy the directory C:\Program Files\Celoxica\PDK\Examples to your "My Projects" directory that you just created. I suggest you change the name to "PDK Examples."

Open and configure your copy of the workspace

In your PDK Examples directory, there are three subdirectories, DSM, PAL, and PSM. Tonight you will work with the workspace under the PAL directory. PAL stands for "Platform Abstraction Layer," which consists of software provided by Celoxica to make it possible to write relatively device-independent code. We will use the same Handel-C files both to generate a simulation program and a downloadable bit file for the RC200E.

Open the *Examples.hw* workspace. You can double-click on it if DK is not running, or you can use the DK File → Open Workspace menu. If you haven't configured Windows Explorer to show file extensions, you won't see the .hw (which stands for Handel-C Workspace) at the end of the file name, but it's there.

In DK, you need to modify the settings for the workspace because the files have been copied from their original locations. The Project → Settings menu will let you do this. In the dialog box that comes up, you will see that you have a choice of which settings you are going to modify. What are the choices? Select "Sim." Look through the various tabs on the right side of the dialog (General, Preprocessor, etc.) Where will the executable code for the simulation program be placed? What preprocessor directives will be defined if you use the Sim configuration? How can a Handel-C program test the values of these directives? The linker tab specifies three library modules that will be linked to the Handel-C program. What are their names, and where are they located in the file system? There is also an Additional C/C++ module that will be linked to the program to generate the software simulation. Change the relative pathname (that starts ..\..\..\…) to the absolute pathname (that starts C:\…) of that library.

Click OK when you have examined the workspace's simulation options and changed that relative path to an absolute one.

Simulate workspace examples

Use the drop down lists in the DK toolbar to select SevenSeg as the Active Project, and Sim as the Build Configuration. Click on the Build icon (the blue one that's two places to the right of the build configuration drop-down list), and the Handel-C program should compile, and generate a simulation program. How many logic gates and how many flip-flops would it take to implement this Handel-C program in hardware? Use the Object Pane to navigate to the seven_seg.hcc source file and look it over, even though it won't make too much sense to you yet.

Click the Go icon, two to the right of the Build icon, to run the simulation. (Or press F5.) Observe that the simulation runs correctly. Stop the simulation by clicking on the Stop Debugging icon on the lower toolbar or from the Debug menu.

Synthesize workspace examples

Switch the Build Configuration to RC200E and build the SevenSeg example. It should fail to build. What tells you that it failed? If you get a message about "Saving bitstream in sevenseg.bit, it means the build did not fail, but follow the directions that follow anyway.)

The problem is that the Build commands for synthesizing the design use relative paths. Use Project → Settings to get the Project Settings dialog box again. Select the RC200E settings, find the Build commands tab, and get rid of the relative pathnames for the commands. These commands are already in your PATH, so you don't even need to give the complete pathname.

Browse to the edifmake_rc200 command file on your computer and look at it with a text editor. What does it do?

Generate the .bit file.

Test workspace examples on an RC200

From the Windows Start menu, find the FTU2 utility under Celoxica → PDK, and start it running. Connect an RC200E to the parallel port of your PC, and download the sevenseg.bit file that you generated in the previous step. Verify that the download succeeded.

Modify workspace examples

Modify the SevenSeg example so that both seven segment displays change in some "interesting" way. Simulate the new design. Synthesize and verify it. Describe the changes you made in your report.

Submit a report of your lab activities

Write answers to the questions asked in this handout, and submit it to me by Monday the 15th. Write your report so that it can be understood without referencing the handout.