# **Microprocessors (662-133) NAME: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Lab #1 – Building a project

Description

This lab will introduce the student to the use and operation of the Nios II EDS software. The student will download a basic Nios II soft processor into the FPGA and then store and run a simple assembly language program.

Learning objectives

1. Create the proper file structure needed by Eclipse.
2. Download the Nios II soft processor into the FPGA.
3. Download an assembly language program into the processor.
4. Use the Nios II debugger to edit and view the program.

Procedure

Logon to the computer.

1. Create a folder to hold your projects (usually named 'workspace'). This may be on a USB flash drive for convenience. If you make it on your system's hard drive, remember:
2. Do not use any folders that have spaces in the names, as some of the Altera tools have issues with this. In particular, Program Files, or My Documents
   1. From Blackboard copy the .sopcinfo file into this folder by right clicking and using “save target as “ <you workspace folder>.
3. Start the NIOS II EDS Software Build Tools (This is a modified version of Eclipse)
4. When EDS starts, it will ask you to choose a workspace. Select the folder you created in step 1. If you like, you can set this as the default, otherwise EDS will ask on every start. Then click ok.
5. Create a new Project by going to File->New-> Nios II Application and BSP from Template.
6. Set the SOPC Information File to the one located in your workspace folder. The CPU name will come up automatically to 'cpu'.
7. Now choose a name for the project. In this case use 'lab1'
8. The Default Location will place all your projects in a subfolder called 'Software'. If you don't wish to have all your labs in the same place, uncheck “Use default location” and change it to simply X:\workspace\lab1
9. For a project template, chose 'Blank Project'
10. Click Next
11. Leave all the settings on this page as default, and click Finish
12. The project will build and after a few seconds, you will see 'lab1' and 'lab1\_bsp' in the Project Explorer. If you see any errors pop up during this step, most likely it is because you used a folder name with a space in it, or you don't have write permissions to the folder.
13. Expand the two folders in the Project explorer using the small arrows to the left of the folder icons. Double-Clicking on any files will open them for editing. Don't make any changes, but explore some of the files automatically generated. In particular 'system.h' in lab1\_bsp will be useful in the future. It contains information about the processor we will use on the FPGA.
14. Let’s create a simple test application:

Right-click on the folder 'lab1' in the Project Explorer and select New->Source File

Set Source file: to 'lab1.s' and change the template to <None>

Click Finish

A window will appear, indicating that the makefiles are being updated, you can check the option to 'do this in the background' from now on, before it closes. A new blank file is created, go to black board and open the lab1.s file by double clicking on it. Highlight the code and then copy and paste it into the blank file in NIOS. After pasting the code into the blank file create a break point in the program by double clicking to the left of the “main” statement. A small blue dot will then appear.

1. Save the file. Then go to Project → Build All. This may take a minute or two the first time.
2. Now we will run this program on the DE1 hardware. Make sure the board is ON, by pressing down on the red power switch and the USB cable is connected.
3. Now we need to setup the debugger. Click the small downward arrow next to the debugging icon
4. Choose Debug Configurations
5. Select the Nios II Hardware option in the list on the left, then click on the New Launch Configuration icon just above. You can change the Name at the top to 'Lab1 Debug' or similar
6. On the first Project tab, change the project name to Lab1, the Project ELF file name: should be automatically filled out as something like X:\workspace\lab1\lab1.elf
7. Go to the Target Connection tab and click Refresh Connections twice on the far right. The USB-Blaster should show up on both lists. You may have maximize your window to see this.
8. Go to the Debugger tab. You can see the option 'Stop on startup at: main' uncheck that box.
9. Click Apply to save changes, then go ahead and click Debug
10. After a few seconds, the project will be built and downloaded to the board.
11. You will see a warning about this kind of launch begin configured to open the Nios II Debug perspective when it suspends. Check the 'Remember my decision' checkbox and click YES
12. Welcome to the NIOS II debugger view.

In the top left, you will see the Debug controls. Don't worry about the information in the window just yet, but make a note of the controls on this window.

You will use these controls to step through programs to observe your instructions in action.

In the middle window you will see your code (if an extra window titled main() appeared, just close it)

1. In the debugging controls click the Step into button, or press F5, this goes through the program one step at a time. You can see the green bar move through each instruction.
2. In the upper right window, select the Registers tab, and expand the main list, then scroll down so you can see r9. Step through the program while trying out the slider switches on the board. What happens to r9? What happens to the red leds?
3. Note that an instruction executes AFTER it is highlighted in green and you press 'step into' so whenever execution has stopped either because it hits a breakpoint (you'll learn about these later) or as you are stepping through, the green highlighted instruction is the NEXT thing that will execute.
4. Press the red Terminate button (or press Control-F2). Program execution stops and the debug window will be blank, and the register view will disappear.
5. To return to the normal programing view, look in the top right where you will see 'Nios II Debug' (it may be slightly cut-off). To the left of it, you will see a thin white line, you can click and drag it to the left with your mouse until you can clearly see both 'Nios II Debug' and 'Nios II'.
6. Click on Nios II to return to the normal programming view.
7. Note always use the drop-down arrow along the side of the debug icon to select the configuration.
8. If it takes a long time to start debugging after you click the 'Debug' button, it may be because of a slow hard drive (since a number of files have to be recreated). If you wish to use a usb flash drive to store your projects, you should use a higher quality / higher performance model to speed up build times.

Note: When you are in the NIOS II debug view, you can make changes to your program and run it again without returning to the NIOS II project view by following these steps.

1. Terminate the program using the red stop button.
2. Make the changes you want.
3. Save the file (you can tell if your file is not saved as there will be an \* by it).
4. Rerun your program by clicking the down arrow next to the debug icon and selecting the debug configuration set up (such as Lab1 debug). Do not simply click the debug icon.
5. Try changing the value on the add instruction from 1 to 2 or another value.
6. Try changing the “addi r9,r9,1” instruction to “muli r9,r9,4”. What does that do? Why is multiplying by a power of two like shifting to the right.