Education Objective

The educational objective of this demo is to become familiar with assembling and debugging NIOS II assembly language programs in the NIOS II Software Build Tools.

Technical Objective

The technical objective of this laboratory is to design an embedded system for the Nios II processor and DE1-SoC that will execute three different assembly language programs that perform the following functions:

1. Find the largest value in a list of numbers
2. Finds the tens and ones digits of a two-digit decimal number
3. Reads the value of the switches and displays them on the LEDs

Demonstration Procedure

1. Open Quartus II and create a new project named assembly_demo
2. Add the file assembly_demo.vhd, found in MyCourses, to the project
3. Open tools > QSYS
4. Create a system with the following components
    a. Nios II/e processor
    b. On-chip memory > size = 32,768 bytes
    c. 8-bit input PIO
    d. 8-bit output PIO
    e. JTAG Uart
    f. SysId

   You can use lab 1 as a guide, but it is important to be able to create the nios_system on your own, so that building the system does not become a stumbling block on future labs and demos. Your finished system should look like figure 1 below.
5. Save you system as nios_system.qsys
6. Generate the VHDL
7. Return to the Quartus project and add assembly_demo/nios_system/synthesis/nios_system.qip to the project
8. Use Assignments > Import Assignments… to import the pin assignments in the DE1_SoC.qsf file
9. Compile the design
10. Program the DE1_SoC board. LED8 should be blinking if the download was correct.
    a. Examine the imported VHDL to see how LED8 is being driven.
    b. Are there any other features implemented in the VHDL?

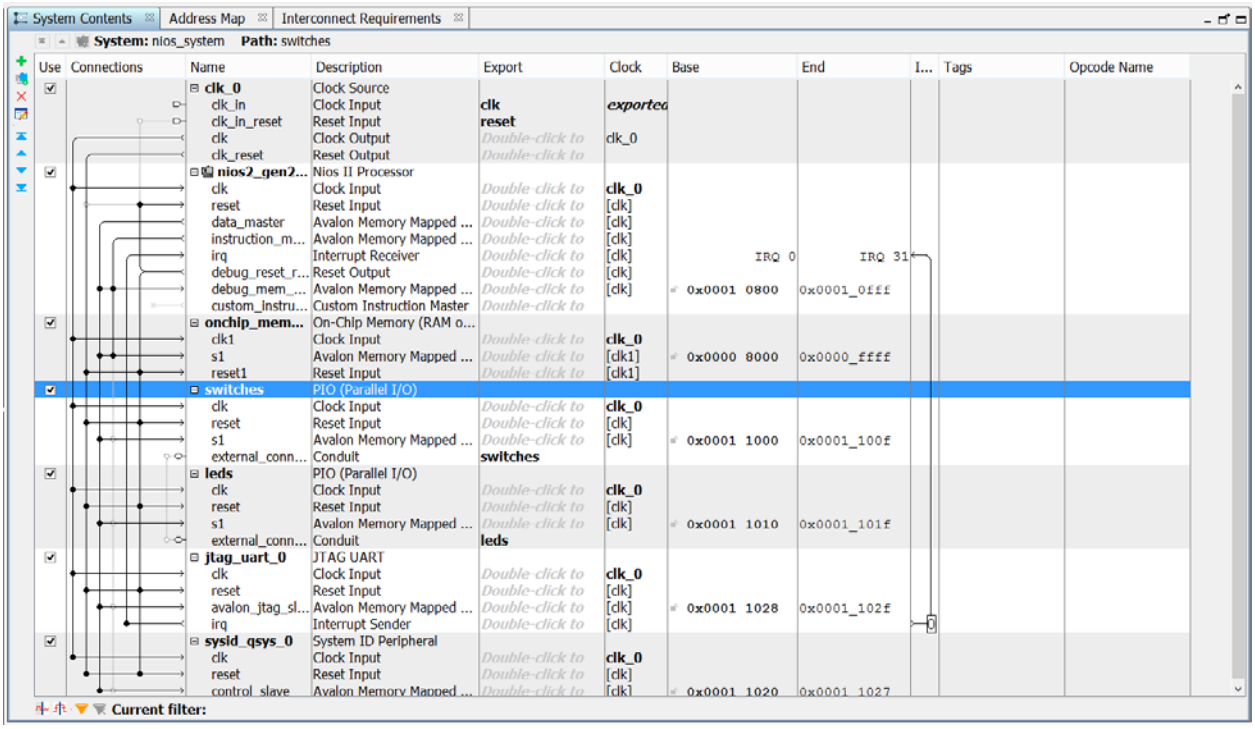
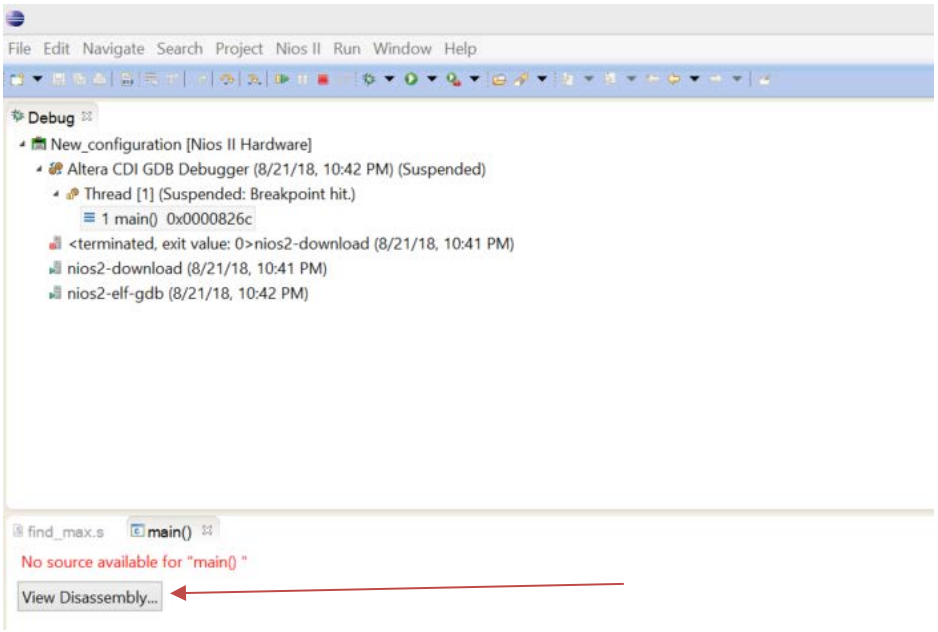
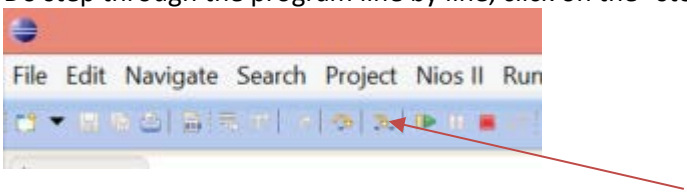


Figure 1. Completed nios_system

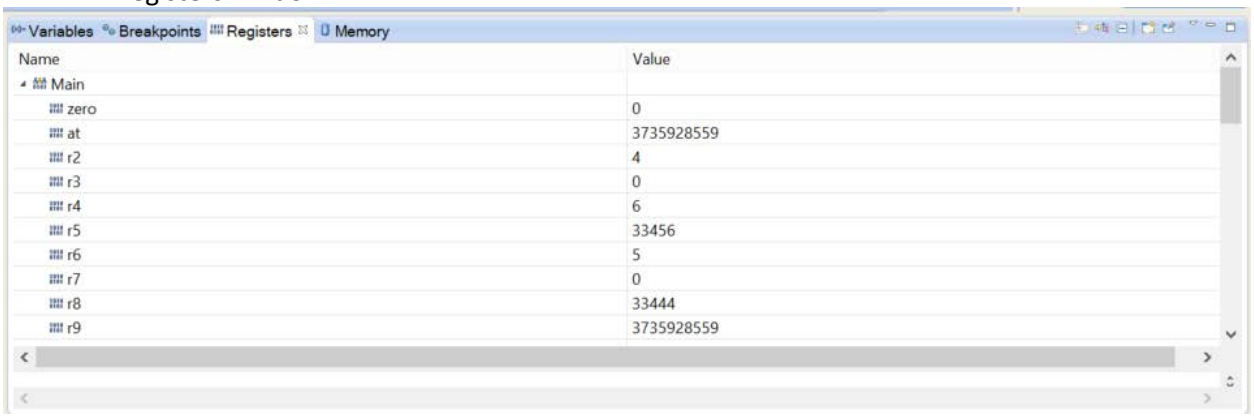
11. Start NIOS II Software Build Tools for Eclipse. Be sure to change the workspace so that it points to the current project you are working on.
12. Choose File > New > NIOS II Application and BSP from template. In the popup box, be sure to choose the SOPC information file for the current project. It will default to the last project you worked on. Choose Blank Project for template.
13. Put the assembly language file called find_max.s in the application folder
14. Right click on the bsp and choose NIOS II > Generate BSP
15. Right click on the bsp and choose Build Project
16. Right click on the application and choose Build Project
17. Choose Run > Debug Configurations. Right click on Nios II Hardware and then New Launch Configuration.
    a. Select project name and *.elf file.
    b. Click on Debug
18. In the Debug Perspective, click on View Dissasembly…



19. Do step through the program line by line, click on the “step into” button



20. As you step through the program, you can watch the values in the register change in the registers window



21. Use the register values to help understand how the program works.
22. Click on the terminate button (red square) to stop execution of the program
23. Remove find_max.s from the project, add digits.s and repeat steps 14 through 23
24. Remove digits.s from the project, add lights.s and repeat steps 14 through 23
25. Open lights.s and system.h. Find the base addresses of the switches and LEDs in the system.h

file and put them in the constants section of lights.s. This allows the program to read from the switches and write to the leds.

26. Repeat steps 14 through 23.
27. Click on the run button and demonstrate proper operation on the DE1-SoC board.

Video Submission:
Show the steps to complete the lights.s portion of this demo. Explain in the video, or in an accompanying text document, what is happening while running the program.