
Education Objective

The educational objective of this laboratory is to investigate the use of the NIOS II assembly language to interface the NIOS II processor with I/O devices on the DE1-SoC development board.

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 an assembly language programs that increments and decrements a single digit counter displayed on the seven-segment display when the pushbutton KEY1 is pressed.

Demonstration Procedure

Part 1 - Hardware

1. Open Quartus II and create a new project
2. Open tools > QSYS
3. Create a system with the following components
 - a. Nios II/e processor
 - b. On-chip memory for program code and data
 - c. 8-bit input PIO for switches
 - d. 4-bit input PIO for pushbuttons
 - e. 7-bit output PIO for hex0
 - f. JTAG Uart
 - g. 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.

4. Save you system as **nios_system.qsys**
5. Generate the VHDL
6. Return to the Quartus project and add **nios_system.qip** to the project
7. Create the top_level VHDL file. The **<project>/nios_system/nios_sytem_inst.vhd** file contains the component declaration and the port map template to instantiate nios_system in the top_level.
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.

Part 2 – Software

1. Write a NIOS II assembly language program that does the following:
 - a. Displays 0 on hex0
 - b. Checks to see if key1 is pushed (active low)
 - i. If SW0 is high, increments the value on hex0
 - ii. If SW0 is low, decrements the value on hex0
 - c. Do not increment or decrement the value on hex0 until key1 is released
2. Below are some suggestions for writing the code
 - a. Use bit masking (learned in microcontrollers) to isolate bits
 - b. Use a subroutine for displaying the current count and for waiting for the key to be released
 - c. Use an array (example in the lecture notes) for storing the seven-segment display constants