Laboratory Exercise 2

Use of Logic Instructions

Logic instructions are needed in many embedded applications. They are useful for manipulation of bit strings and for dealing with data at the bit level where only a few bits may be of special interest. They are essential in dealing with input/output tasks. In this exercise we will consider some typical uses. We will use the DE-series Media Computer with your application programs being loaded in the SDRAM memory.

Since the tasks used in the exercise are of general interest, the students may be able to find various solutions on the Internet. However, in order to maximize the knowledge attained by performing the exercise, we strongly recommend that students develop and test their own solutions.

Part I

Write a Nios II assembly-language program to determine the number of bits that have the value 1 in a 32-bit word. Let the word to be tested be stored in memory location TEST_DATA. Save the result in memory location RESULT.

Create a new project and run your program to demonstrate its correctness.

Part II

Write a Nios II program to reverse the order of bits in a 32-bit word. Let the test word be in memory location TEST_DATA, and place the result in memory location RESULT. Write your program to perform the required task in the minimum amount of time, as measured in terms of the number of clock cycles needed to execute the program.

Create a new project and run your program to demonstrate its correctness.

Part III

In embedded applications it is often desirable to minimize the amount of memory space needed by an application program. Repeat the task in Part II by writing a program that requires as little memory space as possible.

Part IV

Assume that an input device provides an 18-bit signed number, and that this number is stored in memory word INDATA. Assume also that the most-significant 14 bits of this word are set to 0 when the number is read from the input device.

If the stored number is to be used in arithmetic operations, it has to be sign-extended. Write a Nios II program to perform this task. Place the result in memory location RESULT.

Create a new project and run your program to demonstrate its correctness.

Part V

Write a C-language program to determine the number of bits that have the value 1 in a 32-bit word. Use the **scanf** statement to input the pattern to be tested via the terminal window in the Altera Monitor Program. Note that you have to click on the terminal window to activate it for input purposes. Right-click on the terminal window and select Echo input in the drop-down menu, which will display the input characters as you type them on the keyboard. Use the **printf** statement to display the computed result.

Try a number of different test patterns to verify the correctness of your program.

Part VI

Write a C program to reverse the order of bits in a 32-bit word. Use the same input/output approach as in Part V.

Part VII

Write a C program to perform the task in Part IV. Use the same input/output approach as in Part V.

Preparation

As a part of your preparation you should do the following:

1. Write the assembly-language programs for Parts I to IV.
2. Write the C-language programs for Parts V to VII.