Digital Design and Computer Architecture

Lab 1: Full Adder

Introduction

System on chip (SOC) is the current trend in IC design today. With the FPGA, you could design a IP to accelerate the processing progress such as FFT, FIR. The CPU is a general processing unit which could handle any task as long as you could design the code. When compare the two processing units, it is very obvious that the CPU is very good at processing the logic progress, such as processing the keyboard, the input from the mouse and collecting information from the internet and so on. But CPU will introduce huge latency when processing digital signals. On the contrary, FPGA is very good at this but is not good at processing logic progress. So the popular idea is that combining the two part together.

SOC integrate many IPs (for special purposes) and CPU in one chip. The CPU will handle the logic progress and the IP will process the digital signal processing. The DE2-115 board provide a powerful FPGA which contains huge number of gate resources. And the quartus ii and Nios ii software tools provide a way of leveraging FPGA to design a CPU inside the FPGA. So, the basic idea is using the quartus ii to add a MCU (embedded CPU) and the other IPs in the FPGA and then using the Nios ii to write the code to run in MCU.

Requirement:

This lab is not required, and you could just ignore this. This lab provides chances for you to learn some ideas and knowledge about SOC and how to design a SOC system. If you finish and submit report, you could get extra bonus.

The main goal is to design a calculator, using the switches on the board as the keyboard for input and using the 7-segment display as the monitor for output. You can use the 7-segment LED display module in Lab2 as the display module in this project.

Define two sets of the switches as the input of two number. For the input part, you have several implementation methods. You can design a module for collecting the input value on FPGA or you can directly use the MCU to read the switches status for getting the input value. For the input switches, you can use two sets of switches independently or use one

set of switches and input alternatively (first a and then b). For one set of switches, you could add one more switch to indicate whether it is the first input or the second. The input a and b are all one-digit number (0~9).

The operation is "+", "-", "*", "/" (i.e., add, subtract, multiply, division). All these four operation on the two input a,b should be done in the MCU (NOT the FPGA). In another words, you should write C/C++ code to implement the four operation not using the systemverilog.

Based on the requirement, you should find out the output is less than 100, which means you should use two 7-segment displays to show the result. Another thing is that you should use some other switches to indicate what operation you want to do.

This project could be divided into two parts, FPGA and MCU part.

FPGA: Implement the display module using the 7-segment display.

MCU: Run "+","-", "*", "/" (i.e., add, subtract, multiply, division).

The input module is up to you.

Summary: Using the switches to set the "a"," b" and the operation, then, show the computation result using the two 7-segements display. The computation should be done in MCU, programming with C/C++.

What to Turn In

Please turn in each of the following items, clearly marked and in the following order as a single pdf file on Sakai:

1. Your lab5 project files (Quartus and Nios).
2. How you design the project and the SOC.

Let us check your performance on DE2 board NO LATER THAN the week of Dec 9.