Homework 3
Design of 16-bit ALU and its implementation in FPGA

ELEC 5200/6200
Fall 2010

September 27, 2010

1 Problem Statement: ALU design

1.1 Part I:

In this section of Homework 3, you will be designing a 16-bit Arithmetic Logic Unit (ALU) with following specifications using VHDL.

1. ALU has two 16-bit inputs operands
2. ALU performs operations such as addition, subtraction, AND & XOR on the two input operands depending on control lines.
3. The operands can be both positive and negative.
4. ALU provides an overflow error when the result of any operation exceeds range of output word.
5. Your design must have the same input and output port names as those in Figure 1

1.2 Part II:

Simulate the above created ALU in ModelSim. To know more about simulating your design with ModelSim, please refer to the "*ModelSim Tutorial*", posted on course website. This tool can be found on the computers in Lab 320 and Lab 310. You can also access this tool remotely using linux servers. Additional information on ModelSim can be found at http://www.eng.auburn.edu/department/ee/mgc/quickvhdl/modelsim.html

What to submit?

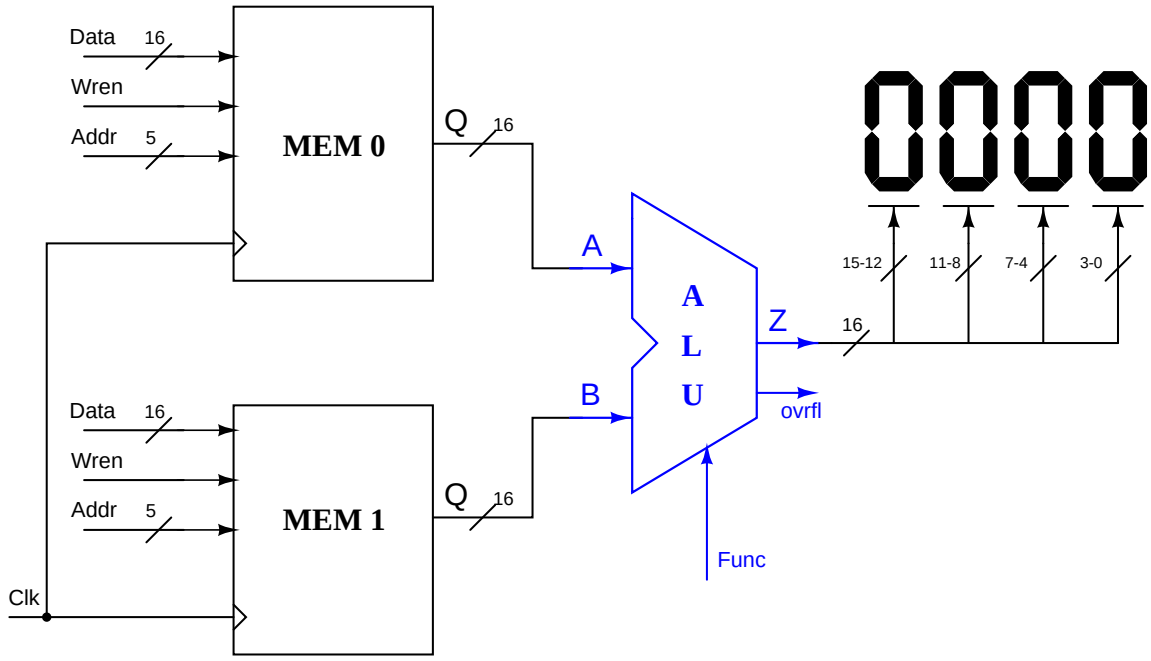

Figure 1: Schematic for testing a 16-bit ALU in FPGA

1. A report which contains a List output of ALU simulations (Do not include your code). Highlight the time instances which verify the functionality of ALU in List output.

2. Comment your code appropriately and email it to TA. Mention your name at the top of your file as a comment.

2 Problem Statement: Implementation in FPGA

Once the ALU design is ready and has been thoroughly tested in ModelSim, we are ready to synthesize and test the design in FPGA. The complete setup of the design required to test the ALU is as shown in Figure 1. You have to design only ALU. Other components in the figure have already been designed and will be provided to you as a wrapper (hw3.zip) on course website. You have to add the above designed ALU in the project folder.

However, it is advised that you go through VHDL description of all other components as well and understand the port mapping in top level design. This understanding is crucial for successfully completing the class project on 'CPU design' which will be assigned shortly.

1. To create a new project and synthesize the design in Altera FPGA, go through "*Altera Quartus II and DE2 manual*". The Quartus II software has been updated recently.

So you might find the screen shots from tutorial a little different from what you are observing on the screen but the contents are the same. After you have created the project, BEFORE YOU COMPILE, go through "*Run time content editable memory tutorial*". Since the memories have already been created and provided to you in hw3.zip folder, you only need to complete step 8 from this tutorial.

2. Also, you need to do following pin assignments by referring to *Altera Quartus II and DE2 manual*

   - clk ⇒ KEY0
   - Addr0[4-0] ⇒ [SW4 - SW0]
   - Addr1[4-0] ⇒ [SW9 - SW5]
   - func[1-0] ⇒ [SW17 - SW16]
   - disp0 ⇒ HEX0
   - disp1 ⇒ HEX1
   - disp2 ⇒ HEX2
   - disp3 ⇒ HEX3
   - ovrfl ⇒ LEDR17

3. Memory initialization files ($Ram\_init.mif$) for both the memories have been provided in the zip folder. If the memories are not being initialized to the values specified in initialization files then mention the complete path name of the file on line number 93 in mem0.vhd and mem1.vhd files. Notice that Altera Quartus II uses forward slashes in the path as against backward slashes used by Windows.

4. How to test the design?

   - Once you have successfully synthesized and downloaded the design in FPGA, the memory now contains data specified in the initialization files. This can be verified by following steps 10, 11, 12 from *Run time content editable memory tutorial*.
   - Now select the addresses for both the memories and choose a function for ALU. Press clock once. You should see the result of the operation on the displays.
   - Verify this result. Apply different functions and operands(various memory words) and check whether ALU functions correctly.

5. Altera DE2 boards can be found in Lab 320. Please go through the above manual carefully before using the boards. After you have successfully tested your design on FPGA, you need to demonstrate it to your TA.