EECS 2021: Computer Organization
Winter 2019

Project (15%)
Simulation of RISC CPU using Verilog Hardware Design

Milestone 1 (5%)

Deadline: midnight Tuesday, 19 March 2019

Upload your group's report in PDF or docx format to Moodle.

ONLY ONE report to be uploaded for each group.

To get the grade for this Milestone, you MUST fill up the following spreadsheet with your group name and group members' names:

https://docs.google.com/spreadsheets/d/1MLJtFLH5GoEZyFvH1iIx8azjhIhVteAGN97VUE8Mwz4/edit?usp=sharing

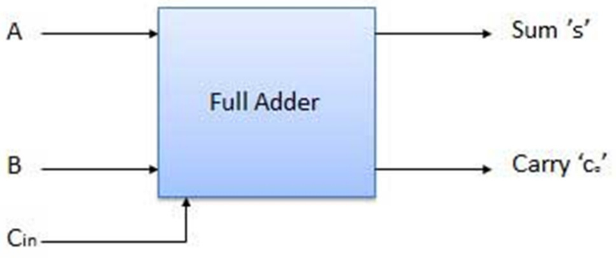
A. Install the Quartus software using the guide provided on Moodle

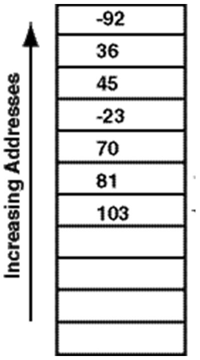
B. Create a new project with simple Verilog code and do the functional simulation of your code using part B and C of the same Quartus guide.

C. Download the zipped project file from Moodle. Unzip and use Quartus to make it into a new project. Compile in Quartus.

D. Write a report about the modules of the Quartus project. Label each section of the report based on the module described inside it. Part A is an overview of the whole project. Part B onwards is about each module/component such as top_cpu, ram.txt and so on. Each part should have a diagram representing it's input/output or representing what it looks like in a computer.

   a. Here are two examples of diagrams (values should be written in hex for easier readability)

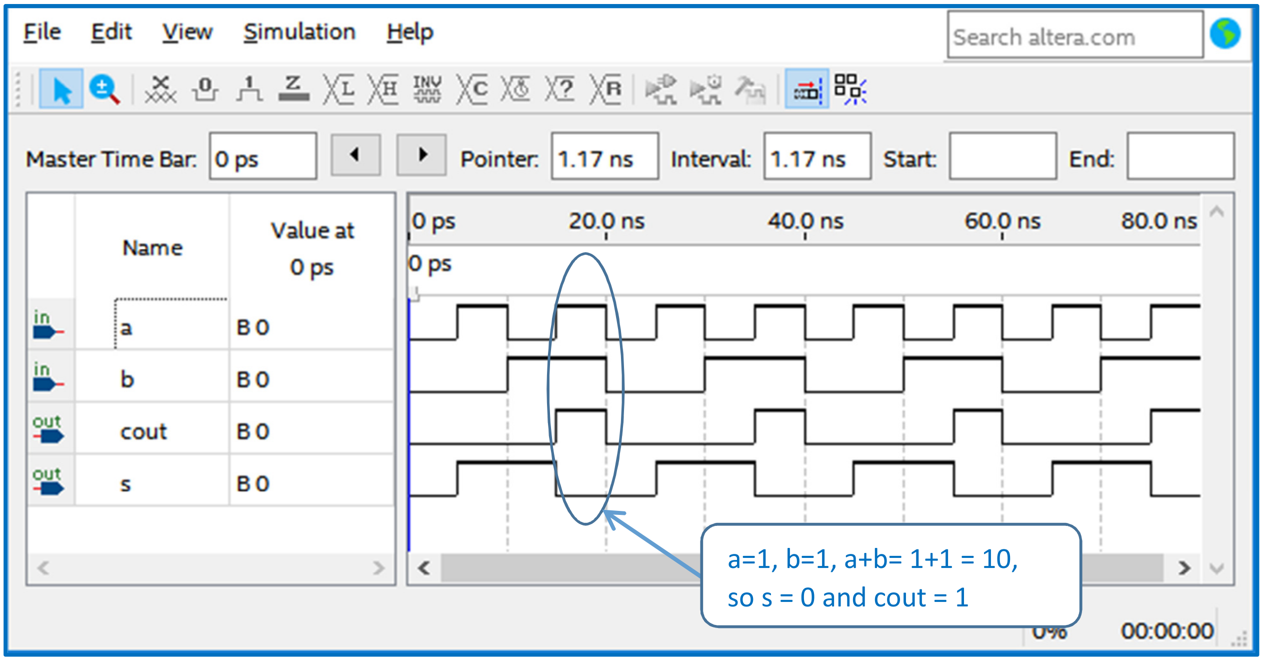




E. In the next section explain how the system works. Break this into as many sections as needed. Your description should have diagrams like a state-diagram, full design diagram that shows connectivity of all modules (You are not allowed to use the diagram generated by Quartus' RTL Netlist Viewer. Lazy students can draw by hand and scan and add to report – BUT it must be neatly drawn ☺).

F. Next section should show the implementation of one or more RISC instructions. Number depends on how many group members there are. For full 4-members, describe two instructions in detail and the rest in shorter descriptions. Description should have an explanation of how a RISC assembly instruction is carried out in this system from start to finish. Where does it get it's operands, where does it send it's result? The description should make it clear you understand how the system works (do you know how many clock cycles it takes to finish an instruction? What role does the FSM play? What are the stages of execution or which module runs first and calls which module? These are sample questions to make you check whether your description is complete. Explain in a different order as long as it makes sense)

G. Show the functionality of the relevant modules by using the functional simulation with University VWF file. Add comments like this example.



a=1, b=1, a+b= 1+1 = 10, so s = 0 and cout = 1

Comment proves that the adder simulation works correctly and explains why this specific input generated this specific output. You should have at least a few different patterns of input to show each module that is simulated is working correctly. For ALU, for example, show at least three ALU operations being done successfully in your simulation. Use three dialogue bubbles in same simulation or use more simulation diagrams if needed.

Add additional description as required. For example, make your simulation output shown in Hexadecimal radix and then add a table that shows binary to hex conversion for the values (such as machine code). Stretch your simulation so that each hex value can be seen in full. Cutoff values = bad simulation = marks deduction. ☹

H. Next section has a table with all members names in one column and the tasks carried out for this milestone by each member in the second column.

Members	Tasks
Michael Jackson	- motivated team members by singing - signed autographs

I. Conclusion section: write what you learned from this project and this milestone.