Yahya Alhinai

EE 4301-LAB

Tuesday

June 14, 2018

LAB 1

VERILOG DESIGN ENTRY, SYNTHESIS, AND BEHAVIORAL SIMULATION

DESCRIPTION OF THIS LAB:

The purpose of this lab was to be exposed to Xilinx Vivado. Using Verilog to describe logic gates, we design a simple circuit. We have learned Synthesis, and Simulation the created circuit. As well as we have learned how to do simple behavioral simulation of Verilog designs.

DISCUSSION OF RESULTS:

We wrote a Verilog code to model an 8-bit full adder using 8 modules compute the results of two 8-bits numbers. The synthesis of the our designed was successful. The 8-bit full adder schematic generated by Verilog code is shown below (figure 1) As well as the logic design for the building block of the full adder modular (figure 2).

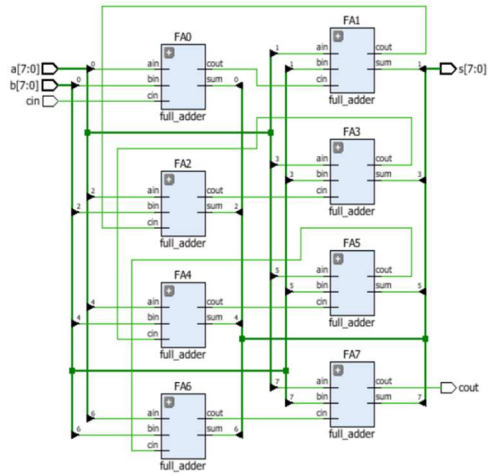

Figure 1: 8-bit full adder schematic generated by Verilog code

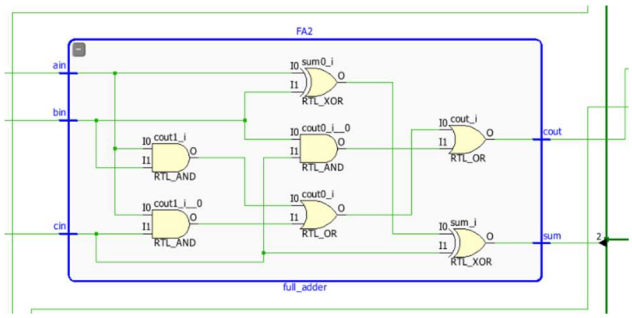

Figure 2: logic design for the building block of a single full adder modular

After making sure that the circuit has no syntax error and it working it meant to be, behavioral simulation of Verilog designs has been made. Different test cases had been simulating to make sure the circuit is fully functioning including overflow triggering. For instance, (figure 3) shows an example of adding A = 8'hF0 with B = 8'h0F and Cin = 1'h01 to give a result of S = 8'h00 and Cout = 1'h01 which indicate an overflow have had happened. All of the cases have behaved as it was predicted previously.

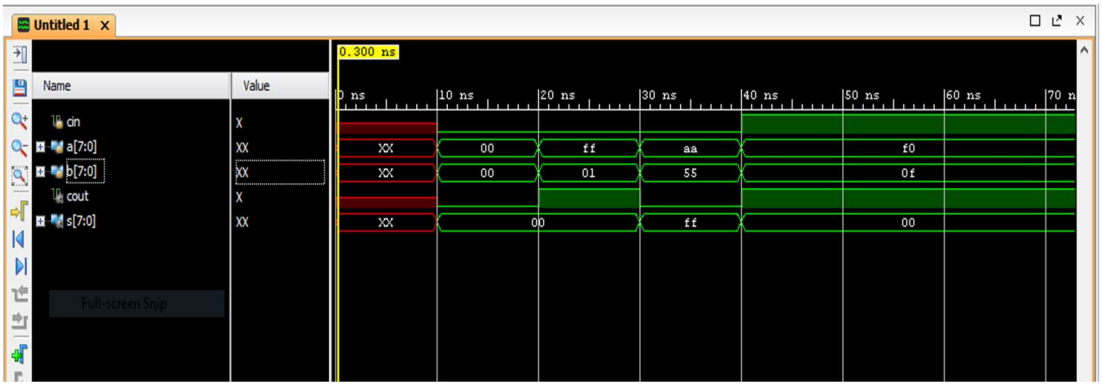

Figure 3: simulated results for the 8-bits full adder

SUMMARY OR CONCLUSION:

All of the predicted results have been met. No difficulties have been faced conducting the 8-bits full adder. There are no prior improvements that can be made.

SOURCE CODE:

The source code that have been created for this and it had been attached separately. The attached files are **full adder modular code**, **8-bits full adder code**, and **a testbench code** to verify the functionally and the behavior of 8-bit full adder.

Wednesday

June 16, 2018

LAB 2

IMPLEMENTATION AND TIMING ANALYSIS

DESCRIPTION OF THIS LAB:

The purpose of this lab was to test different implementation and timing analysis within Xilinx Vivado's environment of the circuit that we have created in the previous lab. We have tried different implementation to test different optimization to the same circuit. We have competed Vivado's default implementation with area explore implementation which have minimize the gates the where used at the expense of power consumption.

DISCUSSION OF RESULTS:

This lab was a continues of what we have started last lab by adding the constrain file and try different optimization of circuit's implementation. The first implementation was Vivado's default. The schematic design on FPGA basys 3 is shown below (figure 1).

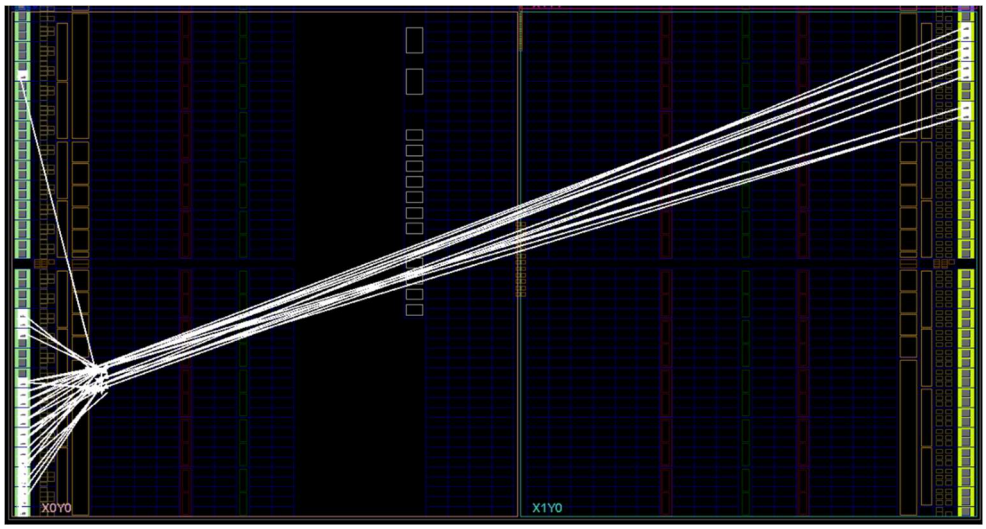

Figure 1: FPGA basys 3 schematic design

Afterwards, area explore implementation had been conducted to mark the difference it and Vivado's default. The two implementations were exactly the same in terms of power consumption and the number of gates used to build up the circuits. This is might be because of the fact that the circuit the was conducted was too simple to be optimized further; therefore, no noticeable difference has been spotted. The power estimation for Vivado's default implementation is shown (figure 2).

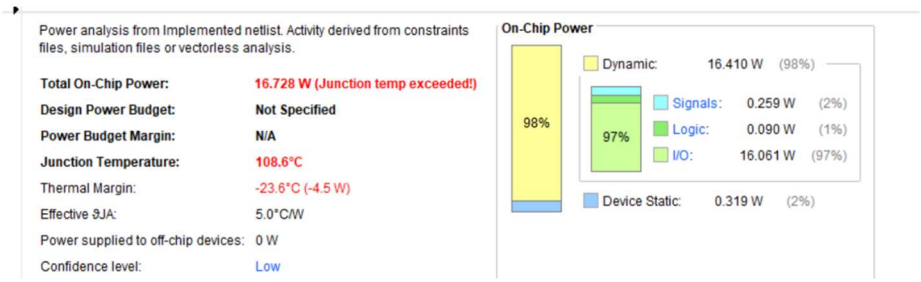

Figure 2: power estimation for Vivado's default implementation

Unlike behavioral simulation which shows ideal transition between different states, timing analysis provide an observation of the delay when switching between two different states. The delay was estimated to be 1.8 ns for the results to be stable. Timing analysis is shown (figure 3). Unlike the prediction that have been made, there is no noticeable difference in timing analysis between the two conducted implementations.

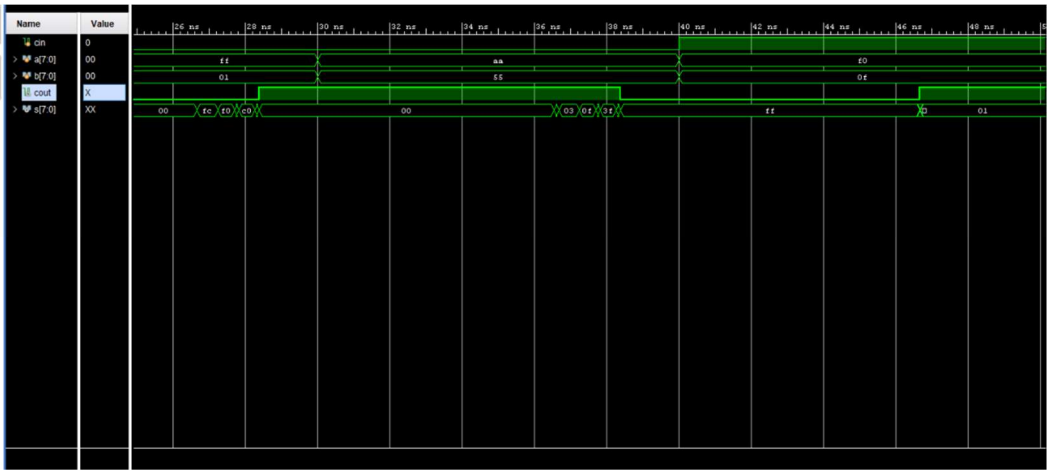

Figure 3: timing analysis for the conducted circuits

SUMMARY OR CONCLUSION:

No difference was observed in power consumption, the number of gates utilizations, and timing analysis unlike predicted results. No difficulties have been faced integrating the constrain file into the project. There are no prior improvements that can be made as far as the lab goes.

SOURCE CODE:

The source code that have been created for this lab and it had been attached separately. The attached files are the same as Lab 1 with further additions. **Full adder modular code**, **8-bits full adder code**, **a testbench code,** and **the constrain file** had been attached.

Tuesday

June 19, 2018

LAB 3

DOWNLOADING TO THE FPGA BOARD

DESCRIPTION OF THIS LAB:

The purpose of this lab was to push a bit stream software file to the basys3 board burring the circuit we designed using Verilog code. We have used the instantaneous programming that erases as soon as the board get restarted. The other type of pushing a code to be preeminently burred into the board by generating a bin file that can be stored in the memory chip on the board. Therefore, every time the board is restarted the writing code in the memory chip kicks in. The design of the full adder from pervious labs had been upgraded to include a full subtractor when a button is pressed.

DISCUSSION OF RESULTS:

In order to add full subtractor modules, I have changed the logic equations of the sum and the carrier to function as a full subtractor. As well as change the constrain file accordingly to accommodate when a button is pressed. When the button is not pressed, the board will compute the given inputs as it activates the full adder modules. When the button is not pressed, however, the two inputs will be subtracted from each other as the full subtractor modules are activated. The results are shown in the board using the 16 LEDs that are in the board. The circuit schematic that implements full adder and full subtractor is shown (figure 1).

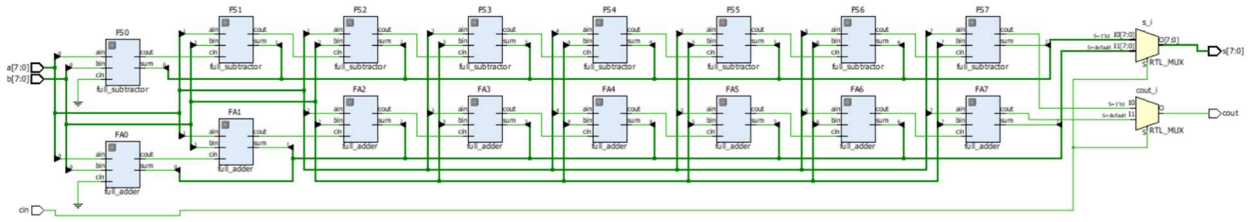

Figure 1: The circuit schematic that implements full adder and full subtractor

After the bin file was uploaded to the board, different inputs have been provided to make sure the full adder and the full subtractor are working as they meant to be.

SUMMARY OR CONCLUSION:

The goals for this lab was fully met. The board behave as it was predicted. No difficulties have been faced pushing the bin file into the board. There are no prior improvements that can be made to improve power consumption or gates utilization.

SOURCE CODE:

The source code that have been created for this lab and it had been attached separately. The attached files are a continuation of previous labs and it has some further additions. **Full adder and subtractor modular code**, **8-bits full adder code**, 8-**bits full subtractor code,** and **the constrain file** had been attached.