**ECE 385**

Spring 2020

Experiment 4

Introduction to SystemVerilog, FPGA, CAD, and 16-bit Adders

Arkajit Dutta, Anand Sunderrajan

Section ABD – Tuesday 6 pm

Nicholas Cebry, Wenjie Pan

**Introduction**

In this lab, we were introduced to SystemVerilog and how to implement code on the DE2-115 board – with the FPGA chip. Compared to the last TTL lab, this was relatively easy as this lab was supposed to ease us into the theory of SystemVerilog, FPGA and other complex topics. We were provided with starter code for the 4 - bit processor and had to implement an 8 - bit processor using the provided code **(explain more about what you did to change the code)**.

For the adders, we were instructed to build three different types of adders – Ripple Carry Adder, Carry Select Adder, Carry Lookahead Adder. The Ripple Carry Adder was built by using a basic adder, where each adder calculates three sums – Carry In, A, and B. The bits are added in each adder and the Carry Out bit from the nth adder is the Carry In bit to the (n+1)th adder. We rippled 16 of these adders together to calculate the final sum. The Carry Select Adder and Carry Lookahead Adder were built using a 4X4 hierarchal structure. The Carry Select Adder is more efficient than the other two adders and it consists of 7 ripple adders, and 3 MUXes. The first ripple adder takes in the first 4 bits of A and B, with Carry in set to 0. The remaining 12 bits of A and B are separated into 4 bits each. Each of the 4 bits have their own logic consisting of 2 ripple adders – one assuming a Carry in bit of 1 and the other assuming a Carry in bit of 0. The sums of these ripple carry adders act as inputs for the 2-1 MUX and the Carry Out bit from the previous acts as the select bit for the MUX. In the Carry Lookahead Adder, we generate two additional signals – Propagate and Generate – in addition to the inputs A and B in order to calculate the carry out bit separately to the sum of each of the inputs and the Carry in bit. The carry-lookahead adder calculates one or more carry bits before the sum, which reduces the wait time to calculate the result of the larger-value bits of the adder.

**8 – bit Logic Processor**

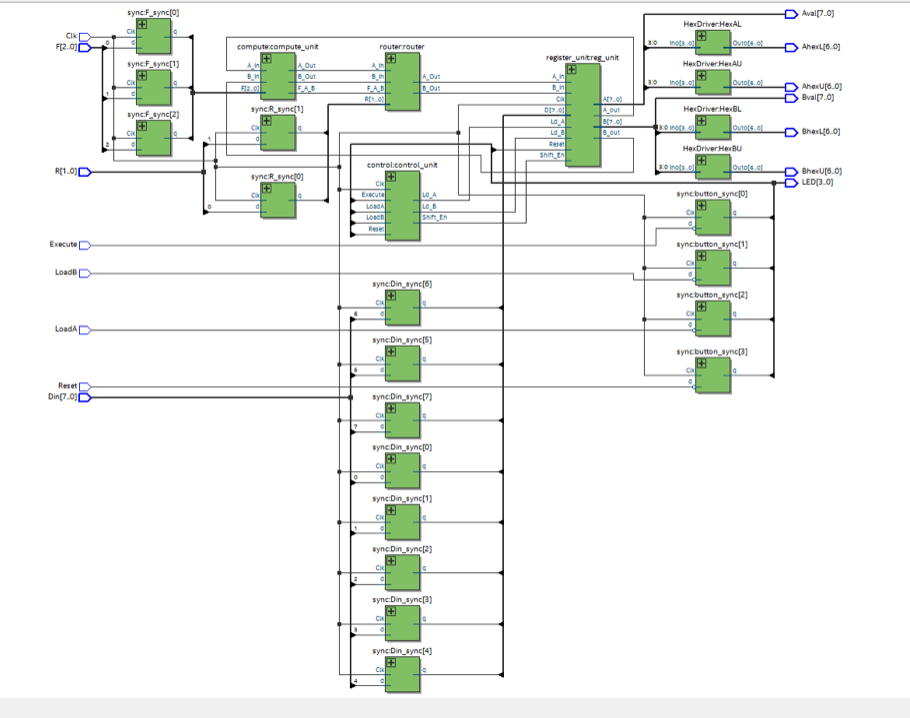


Fig: 8 bit Logic Processor top level

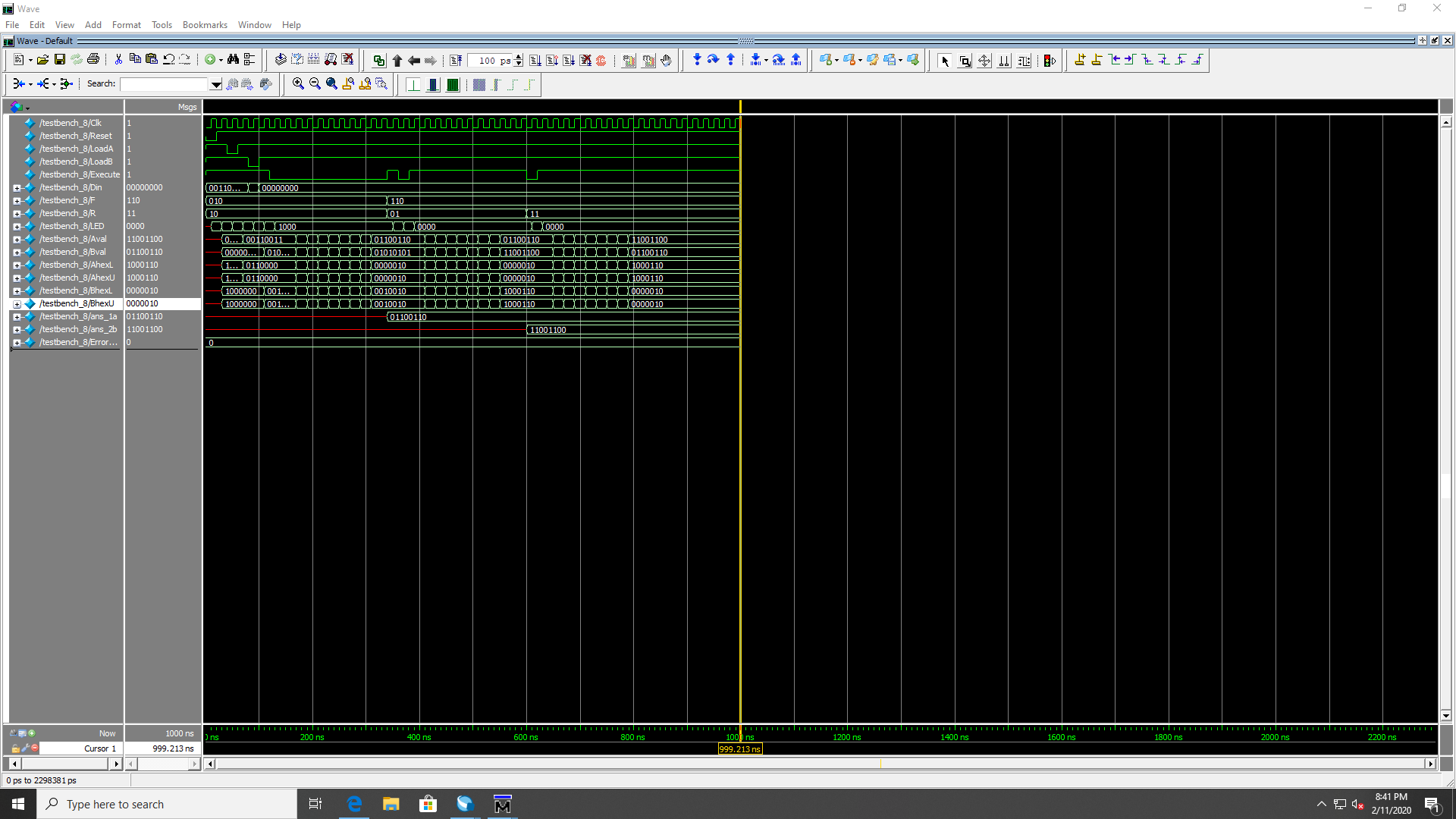


Fig: RTL simulation

**16 – bit Adders:**

1. Ripple Carry Adder:

The Ripple Carry Adder was the easiest to implement as we created a module for a 1 – bit full adder with inputs: A, B, Cin and outputs: Sum, CO. In this module, we calculated the sum of A and B, using the formula Sum = (A ^ B) ^ Cin and the Carry Out bit ( CO ) = A.B