Computer Engineering Program

College of Engineering and Computer Science

Fall 2020



Exercise 3: Carry Look Ahead Adder

EGCP 446 - 10/28/2020

# Prepared by: Levi Randall and James Samawi

Abstract

The Ripple Carry Adder is a staple in many computer engineering applications. Unfortunately, it has one awful drawback and that is time delay. So in more time sensitive cases, the Carry Look Ahead Adder is used for its ability to generate carries and break the chain in the Ripple Carry Adder. In this Exercise, we will be designing a 4 bit Carry Look Ahead Adder using gates to create small modules to be used in the main module.

Summary/Theory

Three modules were designed to create the CLA. The First was a simple Half Adder which was created by putting two inputs (P,Q) into a XOR and an AND gate to create a Carry and a Sum. From here, we instantiate two Half Adders and an OR gate to create a Full Adder with three inputs (U,V,Cin). Finally, we can start making the CLA Adder by setting the Full Adders in a Ripple Carry Adder style without connecting their “Carry In” bits. We now have to design a CLA Unit with only AND and OR gates. This is fairly simple to do when one has the Bit Propagate and Bit Generate Variables, which one can calculate fairly simply by using the equations below.

The carry at any given Full Adder is given by,

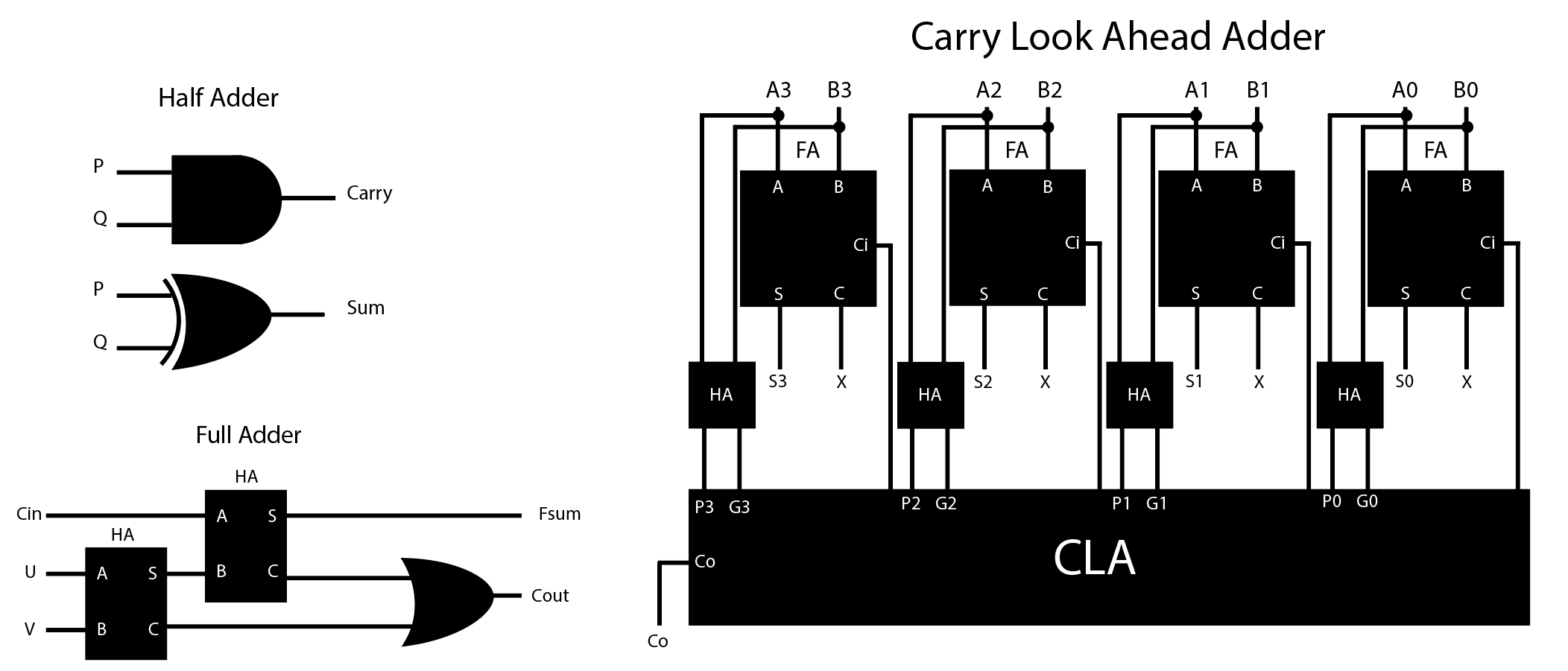
Calculating for the first carry we can see,

And the second,

It then becomes obvious that the only carry needed to calculate the carries for any bit only requires the first bit. Solving for the last two carries,

Using these equations, we can create a logic network to generate all the carries at once and give them to the adders all at the same time. Allowing us to have a very small delay even when calculating numbers with large numbers of carries.

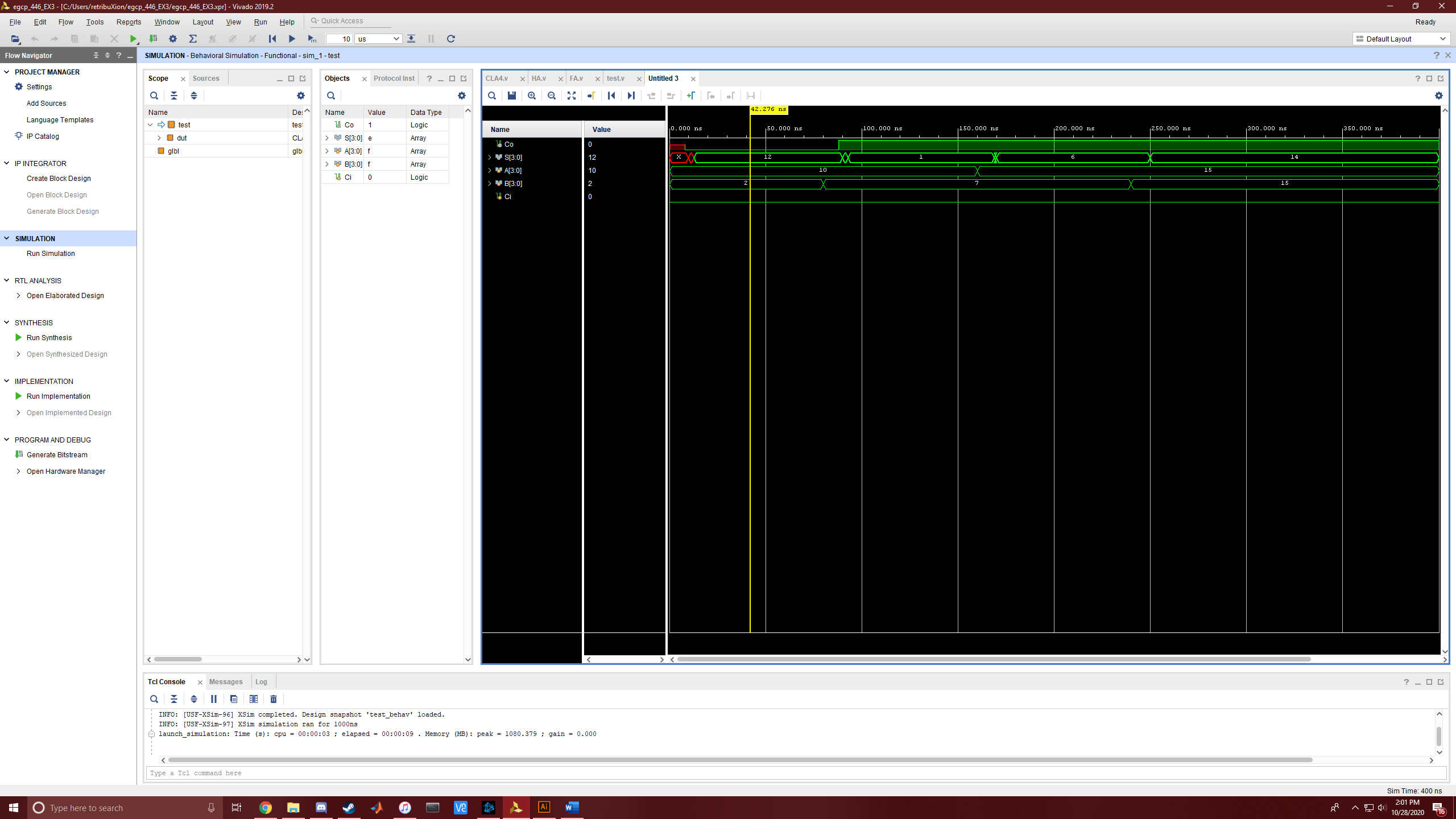
Below is a graphic we created to model our circuits,



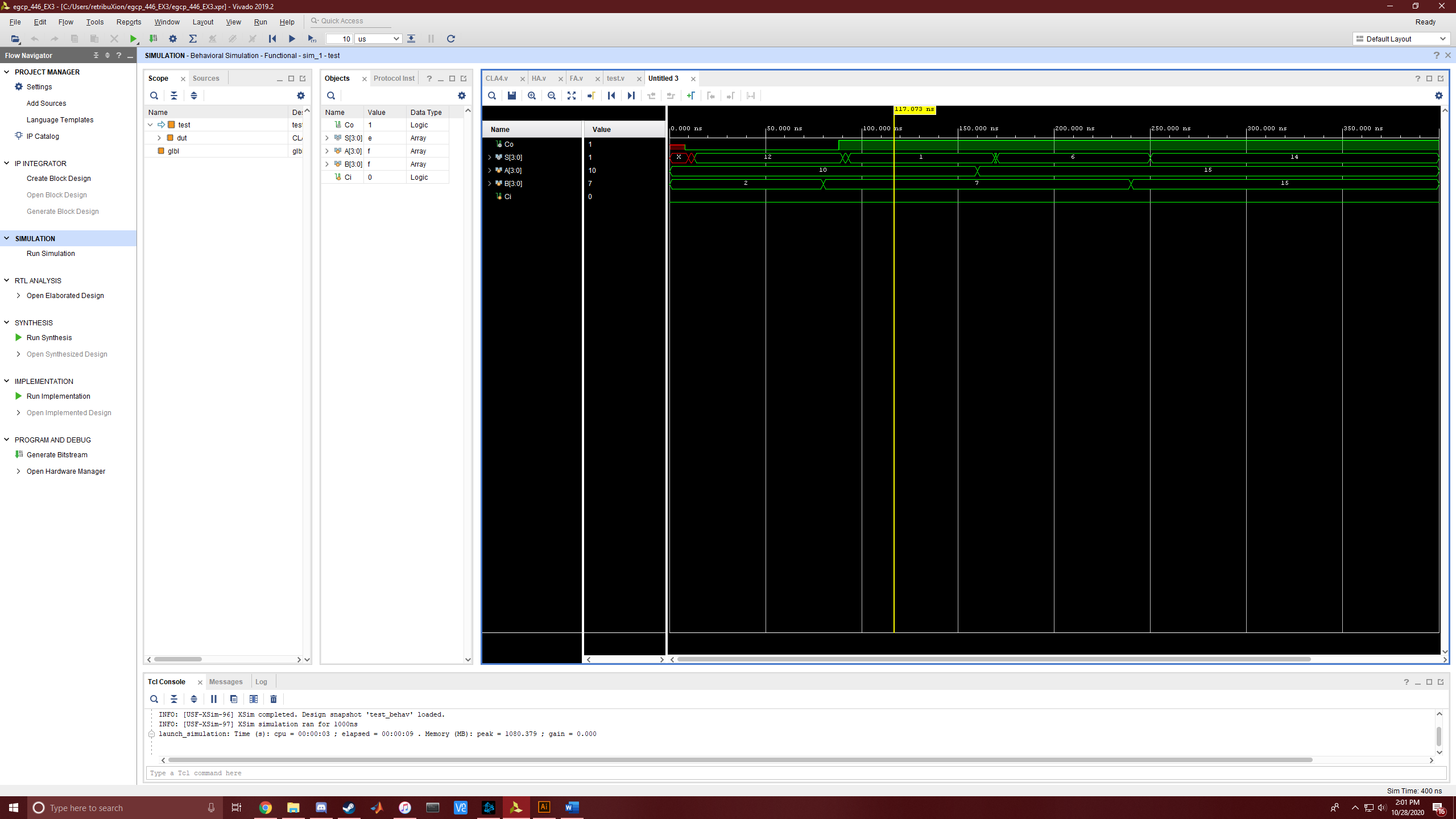
RESULTS

Lastly, we wrote a simple test bench that tested the code when adding different numbers. The first combo was 10 and 2 to make 12. Next was 10 and 7 to make 17. Finally, we added 15 and 15 to see if there was any ripple effect. For all these cases, we got the results we desired and verified our design is functional. No Ripple effect was found and the delay for each result was very low and did not vary with the result entered. The next few screenshots are all images of the results.

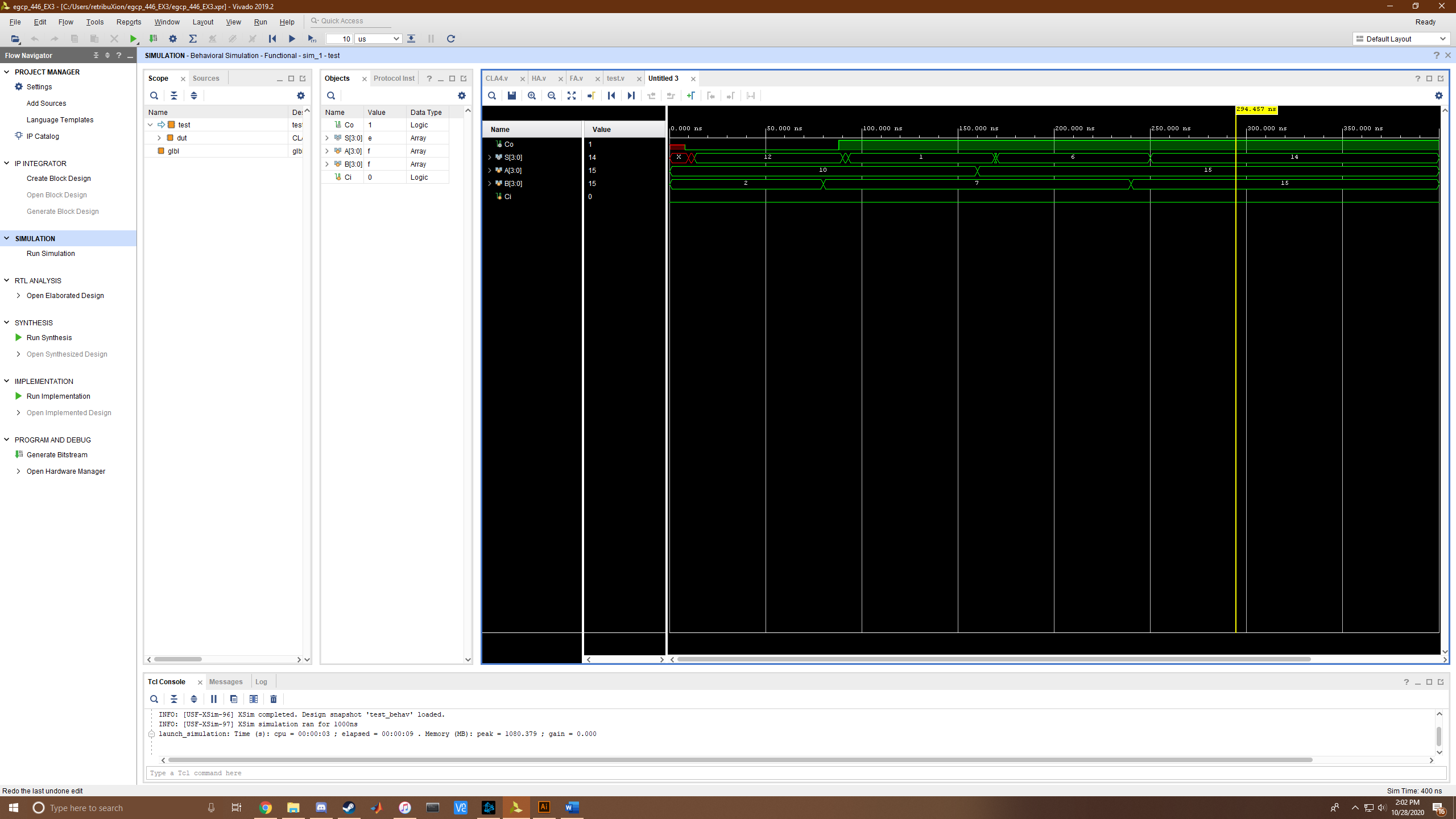
Case 1:



Case 2:



Case 3:



Conclusion

Full Adders are needed in limitless applications. In places where time sensitivity is a constraint the Carry Look Ahead adder works much better than the standard Ripple Carry Adder. Unfortunately, it requires an extra level of design and also requires much more hardware to implement, especially with higher bit values. Overall, the CLA is a good coding exercise at low bit values.