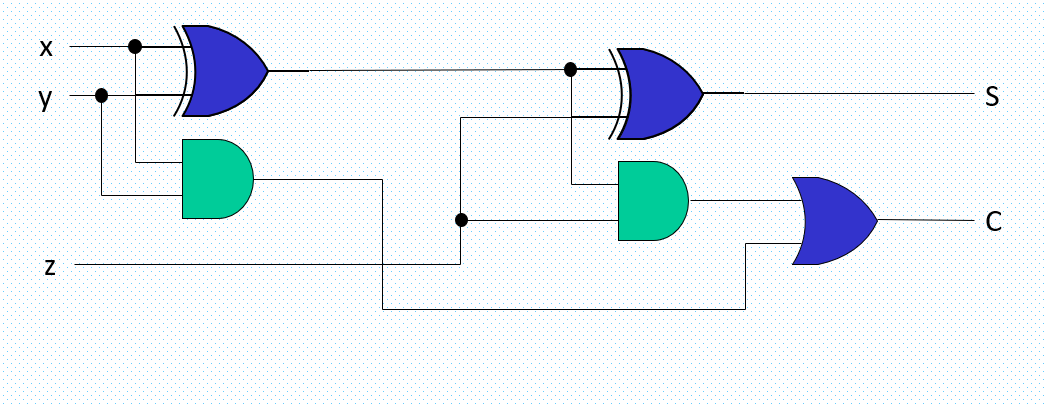
The purpose of this lab was for us to understand the difference between the ripple adders and hybrid adders. The main difference of the two is how they handle the carry. The ripple carry adder waits for each sum to find whether there will be a carry from the summation and then propagate it from least significant bit to the most. Whereas the hybrid carry adder will be looking for a generated carry from the least significant bit to the most again. However even if there is a carry, the hybrid adder does not have to wait for the summation on lower significant bits to accomplish higher significant bits. This should ensures that the time it takes is significantly lower than the ripple carry adder.

However our results were of very small difference in the delay difference (27.966 vs 27.086).

One other difference between the two is the area they use and their expensiveness. The hybrid carry adder uses lower number of gates and occupies a smaller area when compared to the hybrid carry adder.

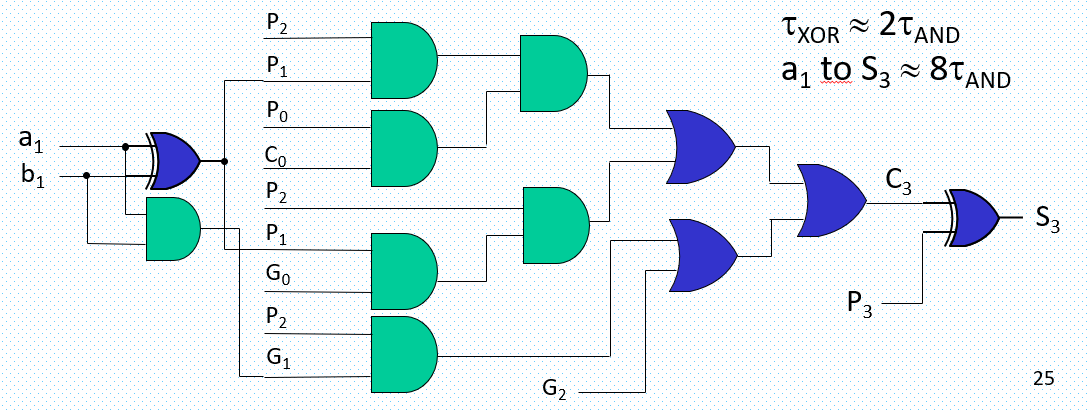
Both of the reports are contained in the zip file.

The way we implemented the ripple carry adder was to implement the full adder once and then call the full adder module in a ripple carry module for 18 times for each bit. We have used XOR OR and AND gates for the implementation as per slides:



1 bit Full Adder

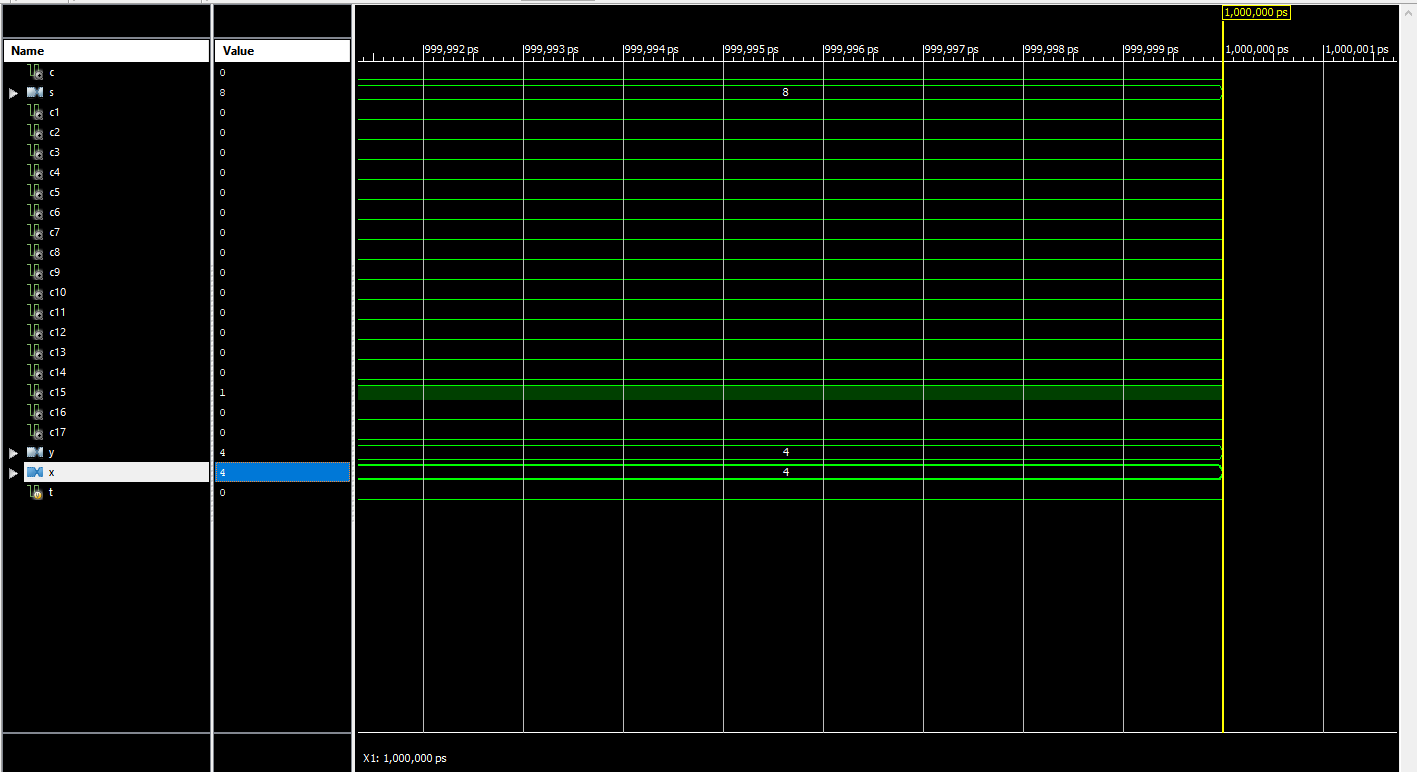
For the hybrid carry adder we have used four 4 bit Carry Lookahead Adders (CLA) and a single 2 bit CLA by implementing one from each. The gates we have used were: XOR AND and OR gates again as per slides:



(only one sum is shown in figure)

Simulation Screen Captures:

Ripple Simulation (contained in zip file):



Hybrid Simulation(Contained in zip file):

