Lab[2] ALU and Carry-Lookahead Adder

To begin this lab, I started to follow the “Hackmd” instructions which I started with assigning the computation of carry bits following the formula of “C[i] = G[i-1] + P[i-1]C[i-1]” for each case and to finish off the 4 bit CLA, I finally calculated the S and the C[out].

Further on, I followed the page 32 ppt slide to module the 16 bit adder with ease. In addition, in the module ALU, I followed the formulas on the “Hackmd” to compute the Y. Later on, to successfully output the module 0100, 0101, 1100 and 1111 , I added a function to compute the add and sub of those modules. The sub adds b’2 complement. Although my problem solving going smoothly, I ran into a deadend which I had no clue of how to the other two whereas after trying to solve my lack of knowledge on the internet which was a no go, I turned into my friends where they gradually explained the codes and fed me with clues to the other implementations. However, after formulating the add, sub, one-hot and find first one from left, I have yet faced another problem with the overflow. I had no clue how to compute it, my study group of friends helped me all the way.

All things in all, on the module 4’d15, I used a while loop but as soon as it finished, the command dc_shell wouldn’t output 1 whereas I had a handful of bugs in the verilog. After thinking for some amount of time, my friends suggested I use casex instead of the while function which worked perfectly except for one flaw. In the end, I couldn’t figure it out but when I ran my program again it magically disappeared.

Cheers,
109006271
Tuguldur Ts.