

Homework Assignment #2**DUE: Fri, Jan 17th**

Consider the equations for a 4 bit carry lookahead adder.

The inputs are B3, B2, B1, B0 A3, A2, A1, A0 and CI (0th index is lsb)

The outputs are SUM3, SUM2, SUM1, SUM0 and C3

Internally you have the signals P3, P2, P1, P0 and G3, G2, G1, G0 and C2, C1, C0

The logic equations are:

$$P3 = A3 \text{ XOR } B3 \quad G3 = A3 B3$$

$$P2 = A2 \text{ XOR } B2 \quad G2 = A2 B2$$

$$P1 = A1 \text{ XOR } B1 \quad G1 = A1 B1$$

$$P0 = A0 \text{ XOR } B0 \quad G0 = A0 B0$$

$$C0 = G0 + P0 CI$$

$$C1 = G1 + P1 G0 + P1 P0 CI$$

$$C2 = G2 + P2 G1 + P2 P1 G0 + P2 P1 P0 CI$$

$$C3 = G3 + P3 G2 + P3 P2 G1 + P3 P2 P1 G0 + P3 P2 P1 P0 CI$$

$$\text{SUM3} = P3 \text{ XOR } C2$$

$$\text{SUM2} = P2 \text{ XOR } C1$$

$$\text{SUM1} = P1 \text{ XOR } C0$$

$$\text{SUM0} = P0 \text{ XOR } CI$$

YOUR TASK:

- 1) Copy file ~degroat/ee762_assign/hw2.vhdl (this is not a web reference!!)
- 2) Edit in your full name in the id block
- 3) Enter (in the appropriate space) a VHDL entity for the 4 bit carry lookahead adder.
NAME the entity CLA
USE TYPE BIT for all inputs and outputs.
- 4) Write an architecture for this entity. Name the architecture DF.
In this problem we want to do a dataflow architecture of the carry lookahead adder so the only statement type that you will use in your architecture is a concurrent signal assignment statement. We will not be using any delay times in this exercise.
- 5) Declare and instantiate this architecture in the architecture of the testbench entity which is below the architecture you just completed.
- 6) Compile the file and fix any errors in your code.
- 7) Simulate the design. Open up a Wave window and add all **signals in design** to the wave window. Run the design until 5200 ns. The easiest way to do this is to enter the command "run 5200 ns" at the prompt in the main simulation window.
- 8) Turn in a copy of your VHDL code and two Waveform printouts from the wave window. On the first printout, print out the simulation of the entire 5200 ns on one page. On the second, print only the results for the first 500 ns.