Create a behavioral dataflow style combinational model for a 4-bit expandable carry lookahead adder along with a testbench to thoroughly verify your model. Recall that a a behavioral dataflow model uses only continuous assignment statements.

Use below Gate delays:

- AND gate 2ns
- OR gate 2ns
- XOR gate 3ns

G (carry generate) is dependent upon current stage's ability to generate a carry.

P (carry propagate) is dependent upon current (and prior) stages' ability to propagate a carry.

A carry is generated at a stage i if it is generated by stage i's inputs (Ai,Bi) or by any prior stage and propagated by every succeeding stage.

Hence,

Cout n+1 = Gn + Cin · Pn

where,
Gn = An.Bn and Pn = An + Bn

Use the design outlined in class. You do not need to use modules for full or half adders. However, note that the design in class did not include a carry in to the LSB. Yours must in order to be expandable.
Using the 4-bit adder module you created and verified, create an expandable 8-bit adder and thoroughly verify it.

Correctness, clarity, reusability, and maintainability count.