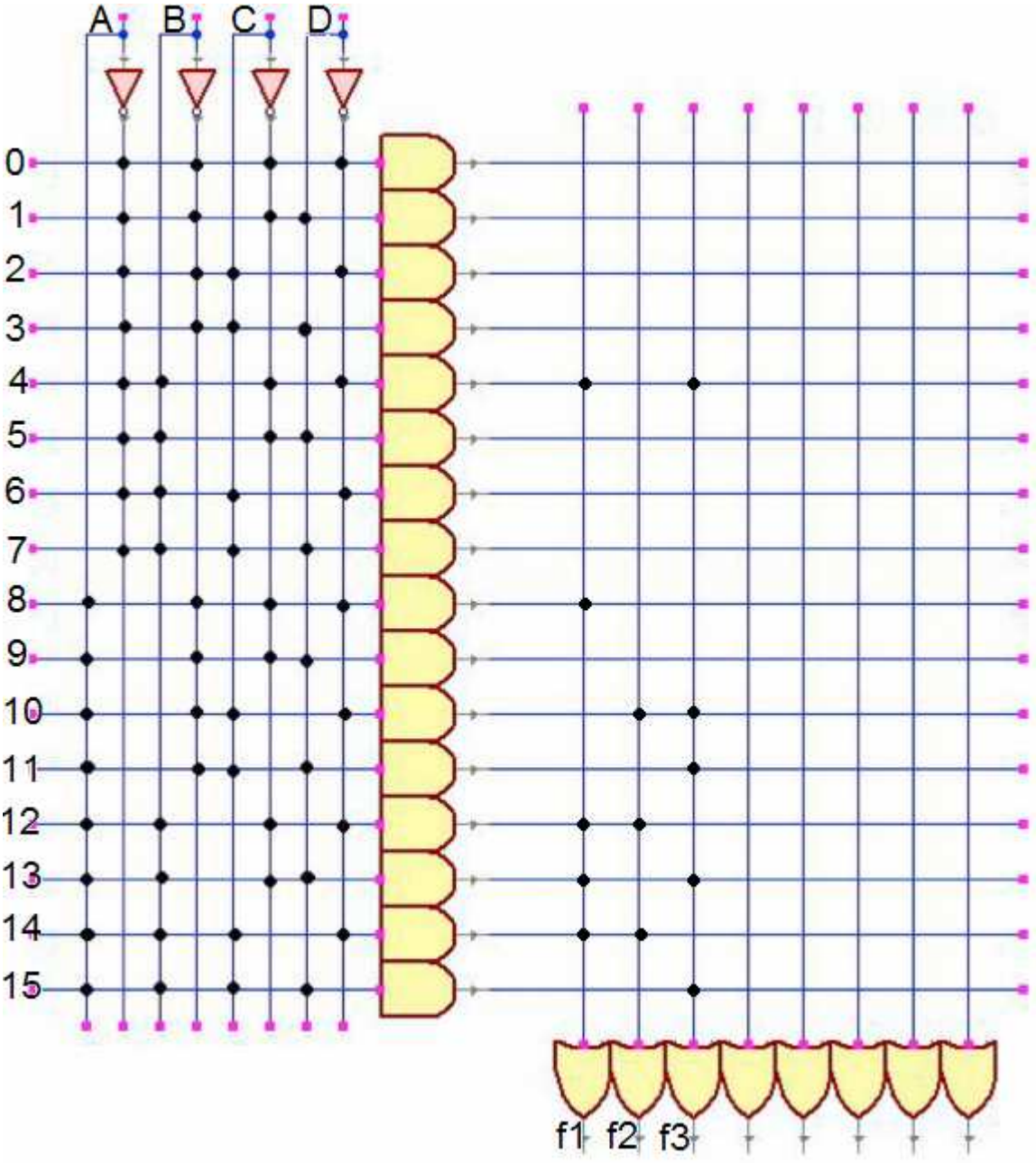
CSE 370 Homework 4 Solutions

1. ROM, PLA, PAL

a) ROM – don't reduce, just fill in the bubbles from the minterms in the problem statement



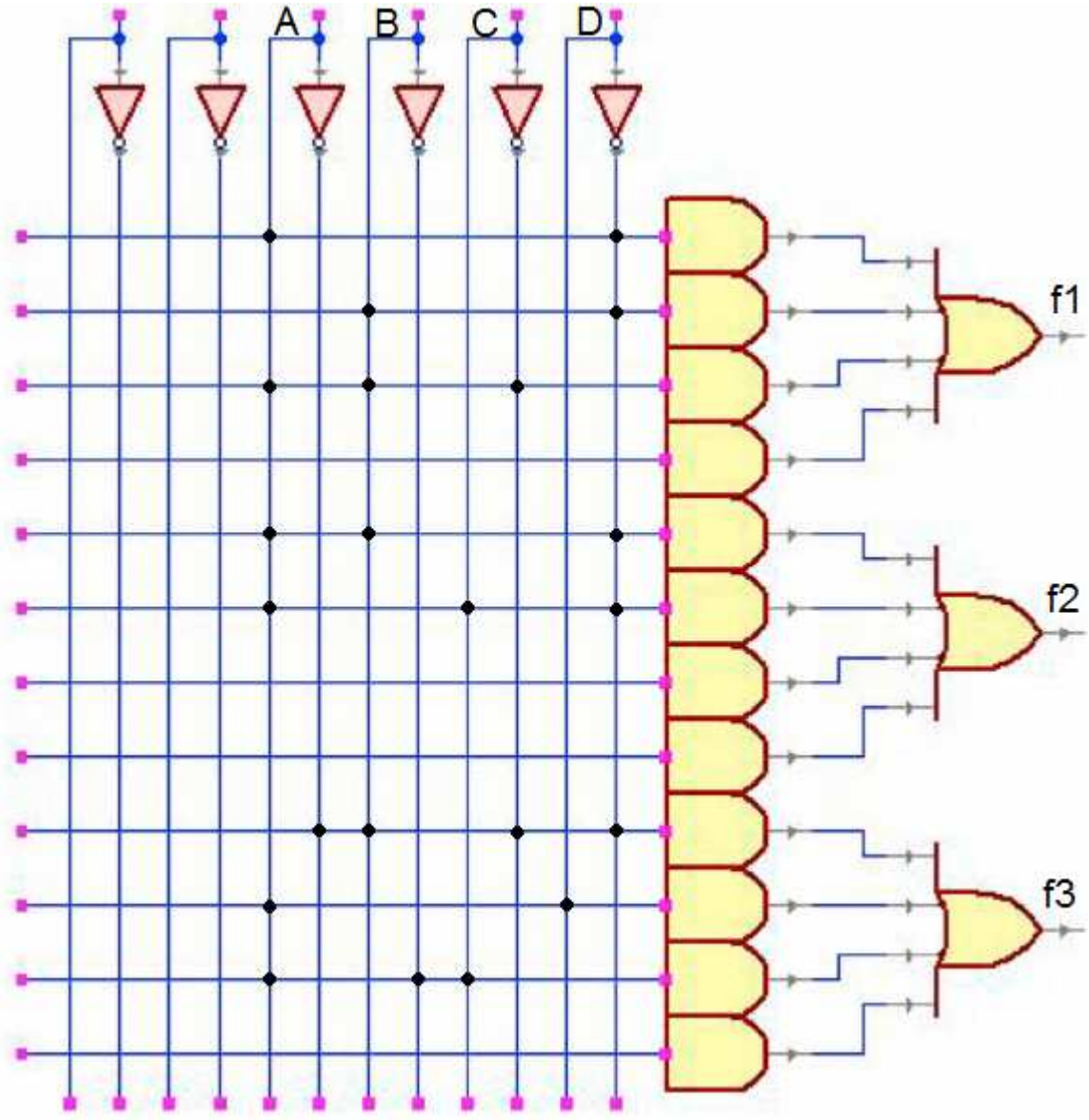
b) PAL

These functions reduce to:

f1 = BD’ + AD’ + ABC’

f2 = ABD’ + ACD’

f3 = A’BC’D’ + AD + AB’C



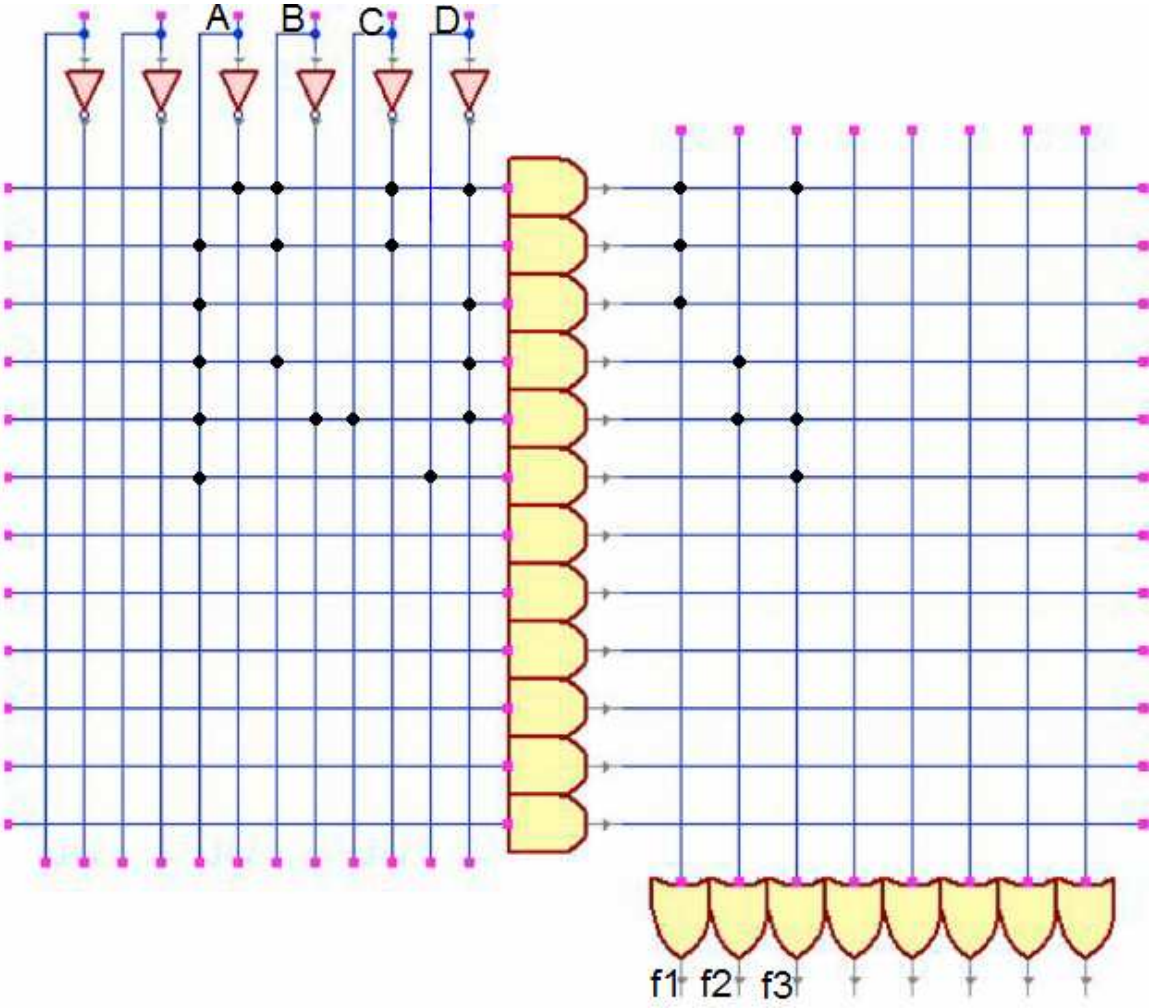
c) PLA

To minimize the number of AND gates, look to reuse AND gates between functions. Doing this gives the following Boolean equations:

f1 = A’BC’D’ + ABC’ + AD’

f2 = ABD’ + AB’CD’

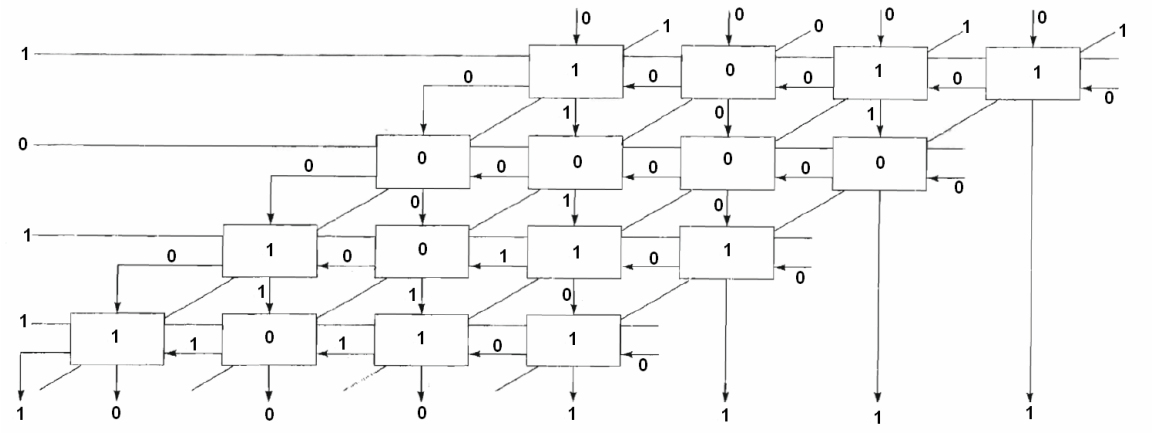
f3 = A’BC’D’ + AD + AB’CD’



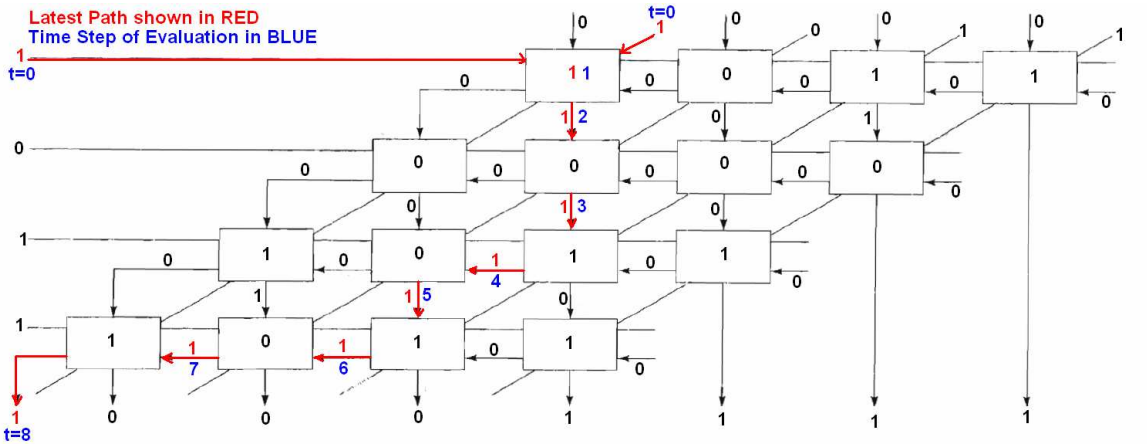
2. Multiplier

All values for the multiplier array shown in the diagram below. Input A is listed along the top diagonals, input B is listed along the side, partial products are shown inside the full adders, carry bits are shown as flowing from the full adders to the left, and sum bits are shown flowing from the full adders downward.

Note: for the timing questions of this problem, assume the generation of the partial products and the calculation of the carry and sum bits all take 1 unit time step. This may not be accurate, due to the different circuits required to realize the different functions, but this assumption greatly simplifies the delay calculations.



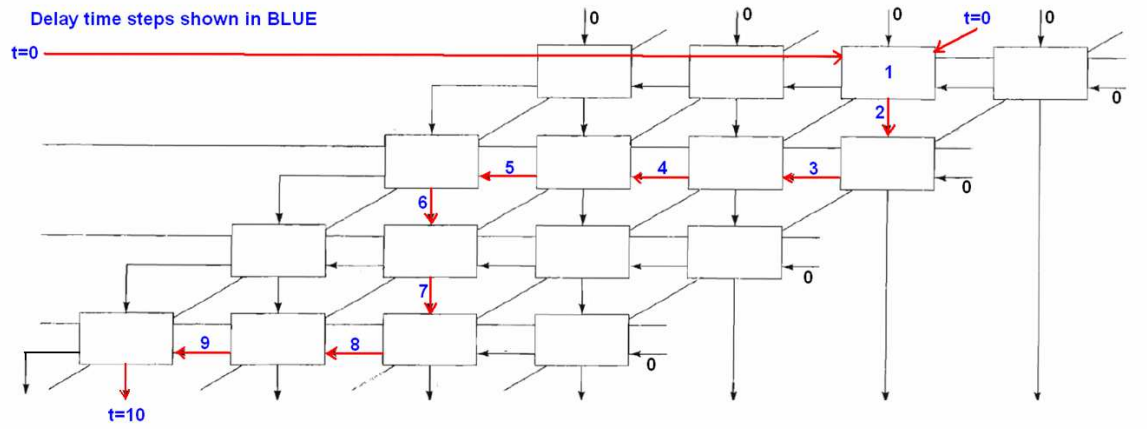
a) Path of slowest output given these inputs is shown in red in figure below



b) Worst case path for this multiplier is shown in red in the figure below.

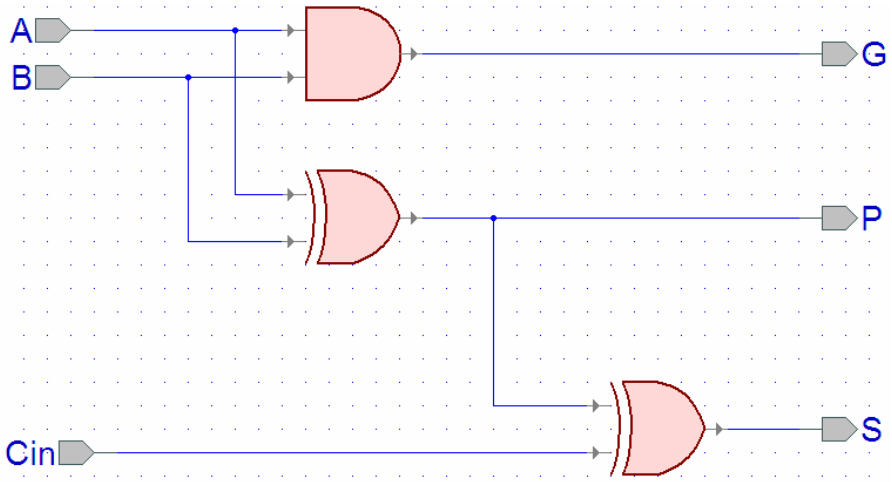
This worst case path is seen when multiplying:

A = 0111 (along the top) and B = 1011 (along the side)



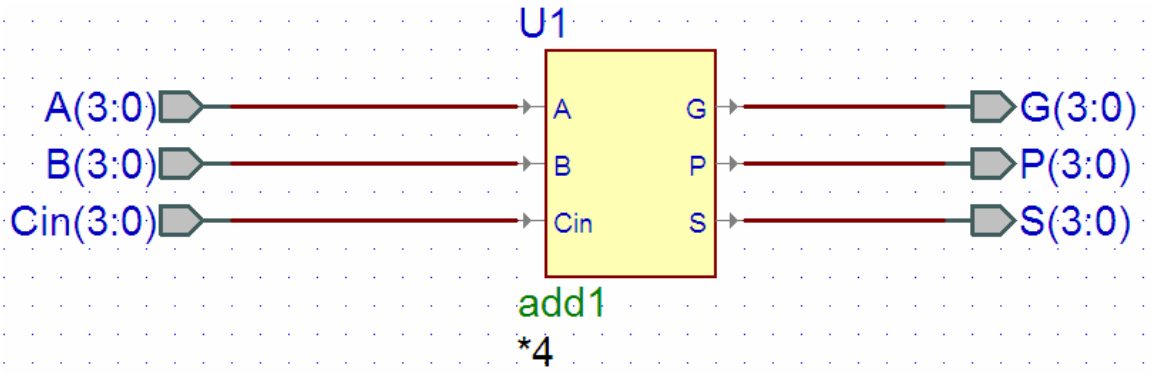
3. 16 bit CLA Adder

a) 1 bit adder – schematic or verilog



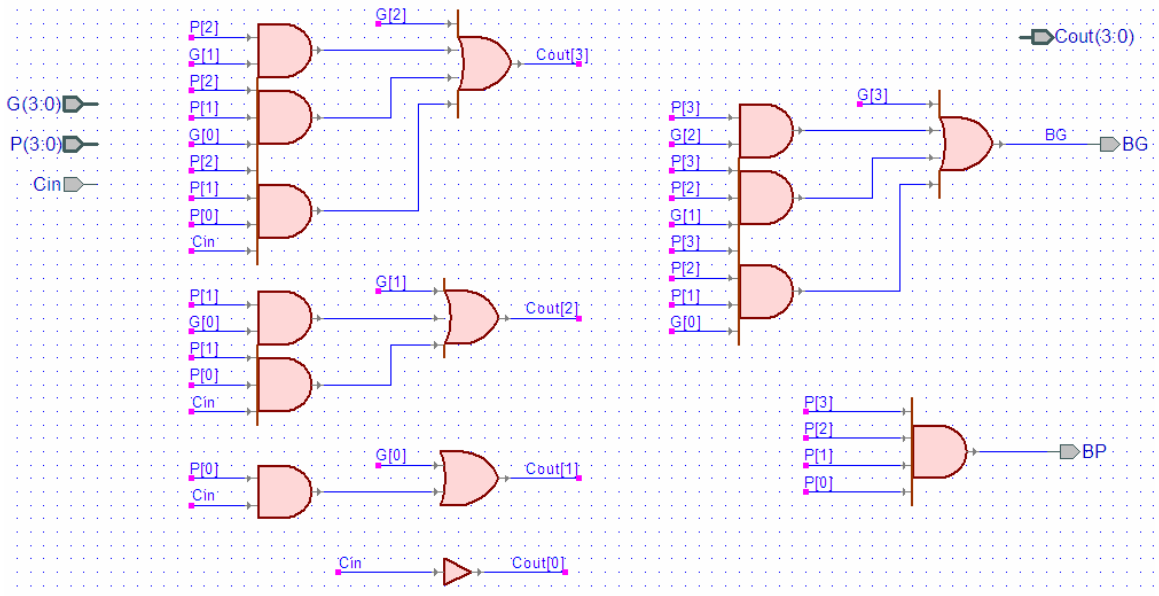
Note: Notice that propagate and generate signals are computed after 1 gate delay, and sum is computed 1 gate delay after the carry in is known.

b) 4 bit adder – schematic only



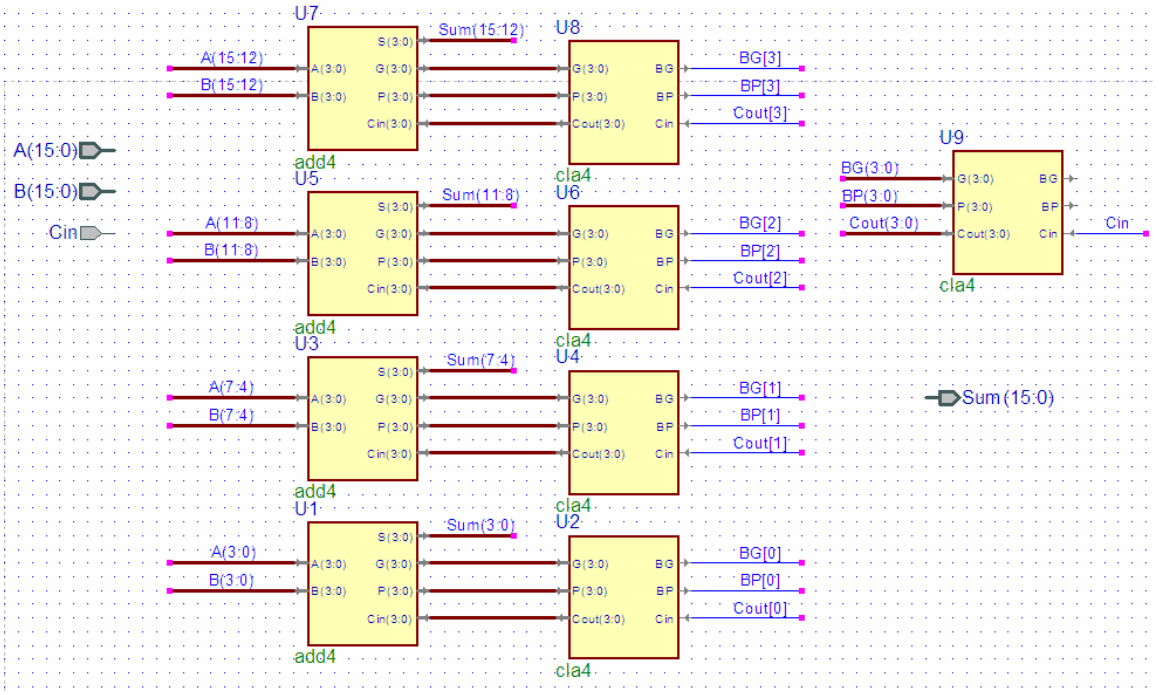
Note: This shows an implementation using vector notation. This could also be done without the vectoring, but the 4 1-bit adders must be independent of each other.

c) 4 bit CLA component – schematic or verilog



Note: BG is block generate, BP is block propagate.

d) 16 bit CLA Adder – schematic only



e) Note: when counting gates and delay, assume buffers are free

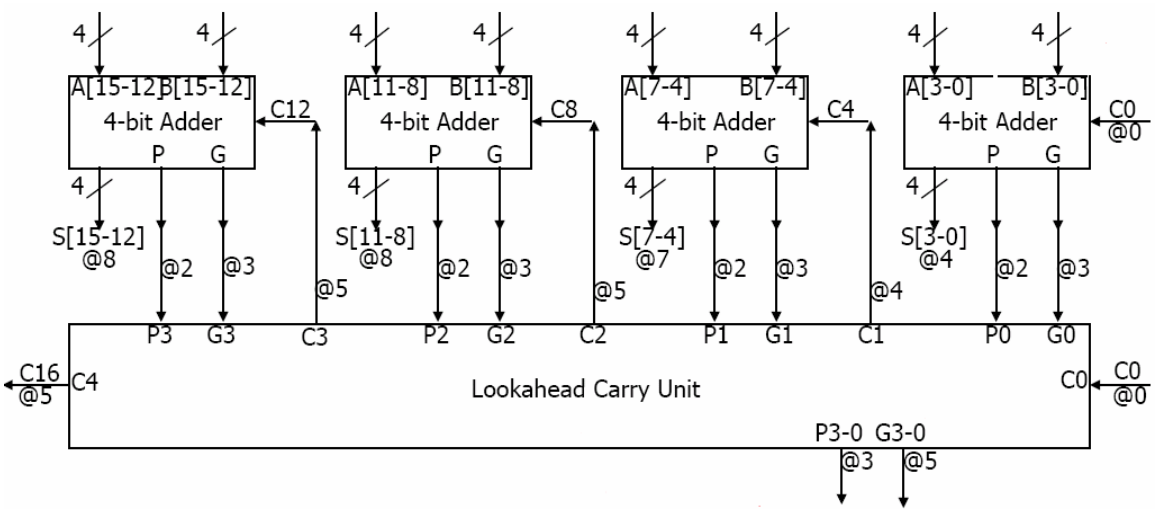
gates = 4 * add4 + CLA_4bit * (4 + 1)

gates = 4 * (4 * add1) + (10*AND + 4*OR) * 5

gates = 16 * (1*AND + 2*XOR) + 14 * 5

gates = 118

delay calculation for 16 bit CLA adder is shown here:



delay = (pi,gi calculation) + (P,G calculation) + (Block carry generation) + (individual carry generation) + (sum calculation)

delay = 1 + 2 + 2 + 2 + 1 = 8 gate delays

notes:

pi = ai ^ bi; gi = ai * bi; = 1 gate delay

P = p3*p2*p1*p0 = 1 gate delay

G = g3 + g2*p3 + g1*p3*p2 + g0*p3*p2*p1 = 2 gate delays

C2 = G1 + G0*P1 = 2 gate delays

ci = gi-1 + gi-2*pi-1 = 2 gate delays

sum = pi ^ ci = 1 gate delay

f) # of gates and delay of 64 bit CLA

gates = 16 * add4 + CLA_4bit * (16 + 4 + 1)

gates = 16 * 12 + 14 * 21

gates = 486

delay calculation shown here:

note: this is very similar to the 16-bit, but another level of calculating the group propagate and generate signals, then the group carry in bits must be added

delay = (pi,gi calculation) + (Block P,G calculation) + (Group P,G calculation) + (Group carry generation) + (Block carry generation) + (individual carry generation) + (sum calculation)

delay = 1 + 2 + 2 + 2 + 2 + 2 + 1 = 12 gate delays