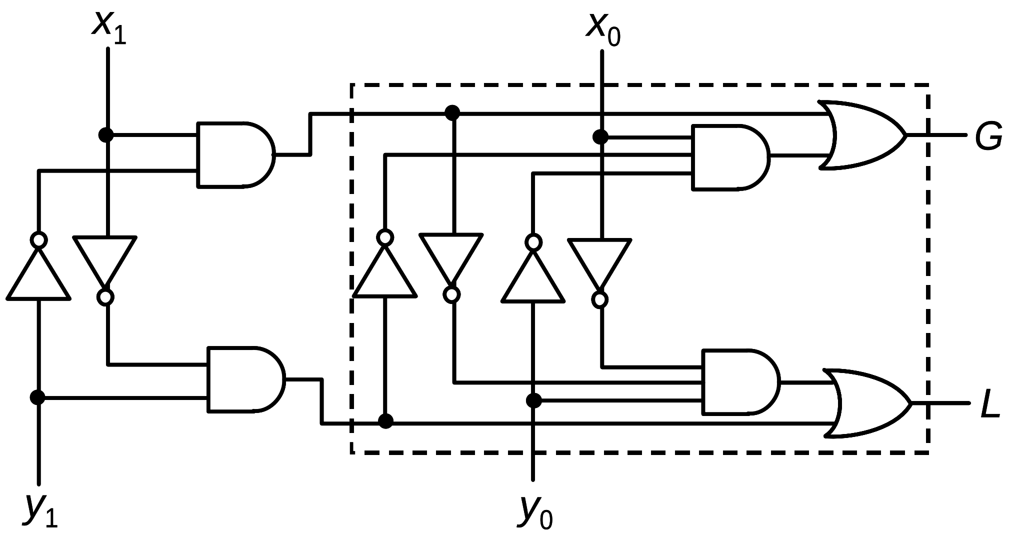
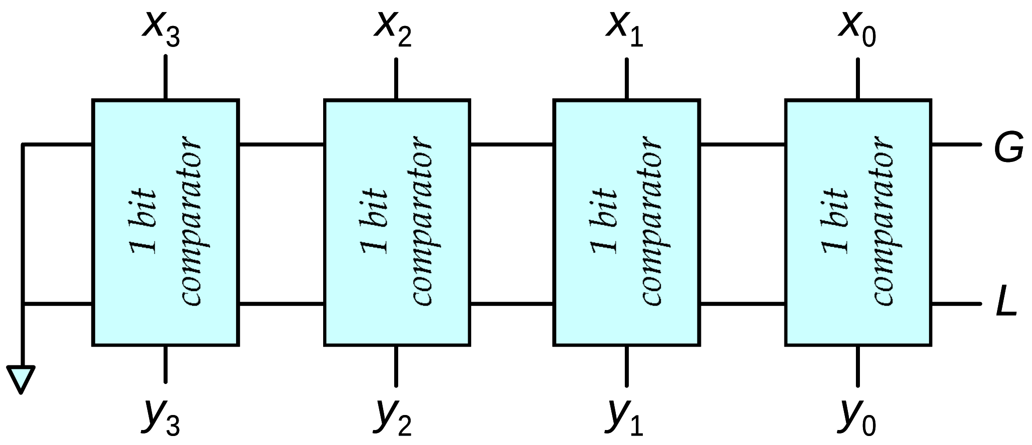
12. An *n* bit comparator takes two *n* bit input values *x* and *y* and has two outputs *L* and *G.* If the value of *x* is numerically smaller than *y*, then *L* is high and *G* is low. If the value of *x* is greater than *y*, then *G* is high and *L* is low. Otherwise, both *L* and *G* are low. An *n* bit comparator can be implemented using *n* 1 bit comparators. Show how to implement a 4 bit comparator in this way. (Note that the outcome of the comparison is determined entirely by the most significant bit position in which the value of *x* differs from *y*.)

The two outputs L and G of a 1 bit comparator are defined by the logic equations $L=x'y$ *and* $G=xy'$*. For a 2 bit comparator, the corresponding equations are* $L = x_1'y_1 + (x_1y_1')'(x_0'y_0)$ *and* $G = x_1y_1' + (x_1'y_1)'(x_0y_0')$*. This leads to a circuit that looks like this.*

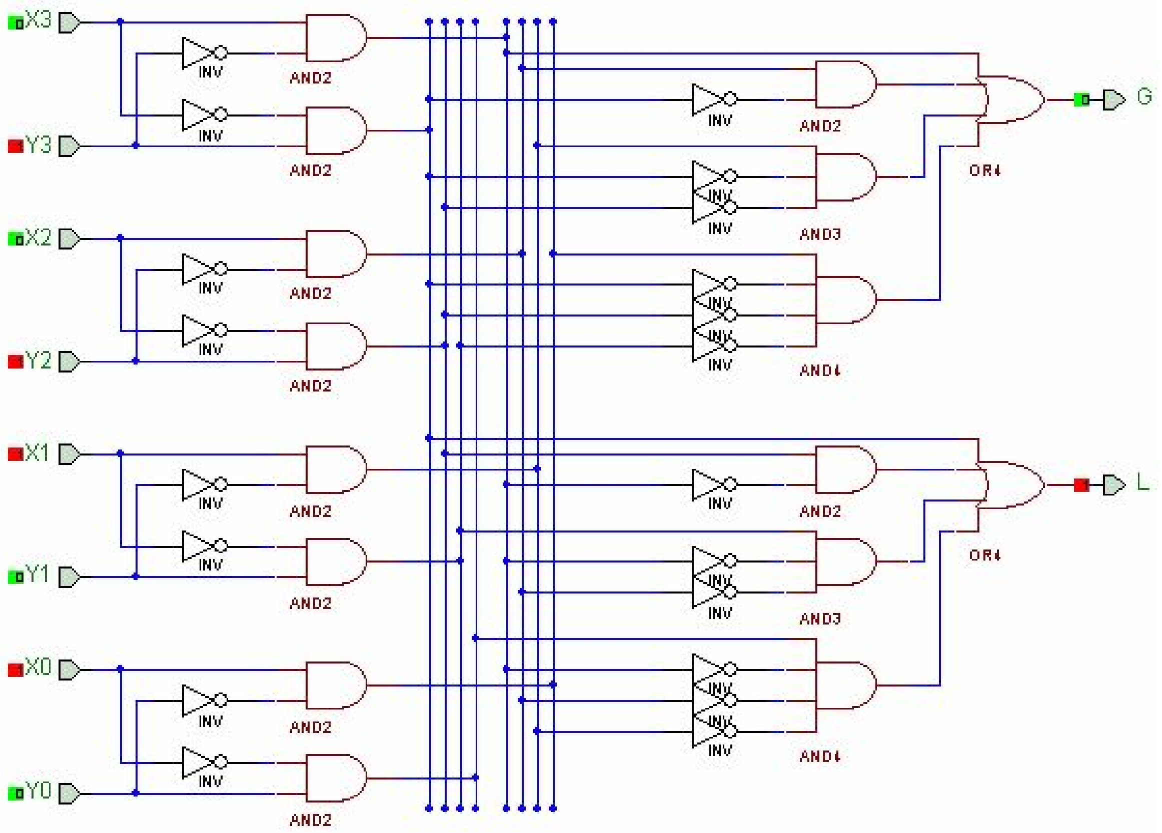


The portion of the circuit highlighted by the dashed lines can be cascaded to produce an n bit comparator. So, for example, using 4 copies of this circuit, we can get the 4 bit comparator shown below.

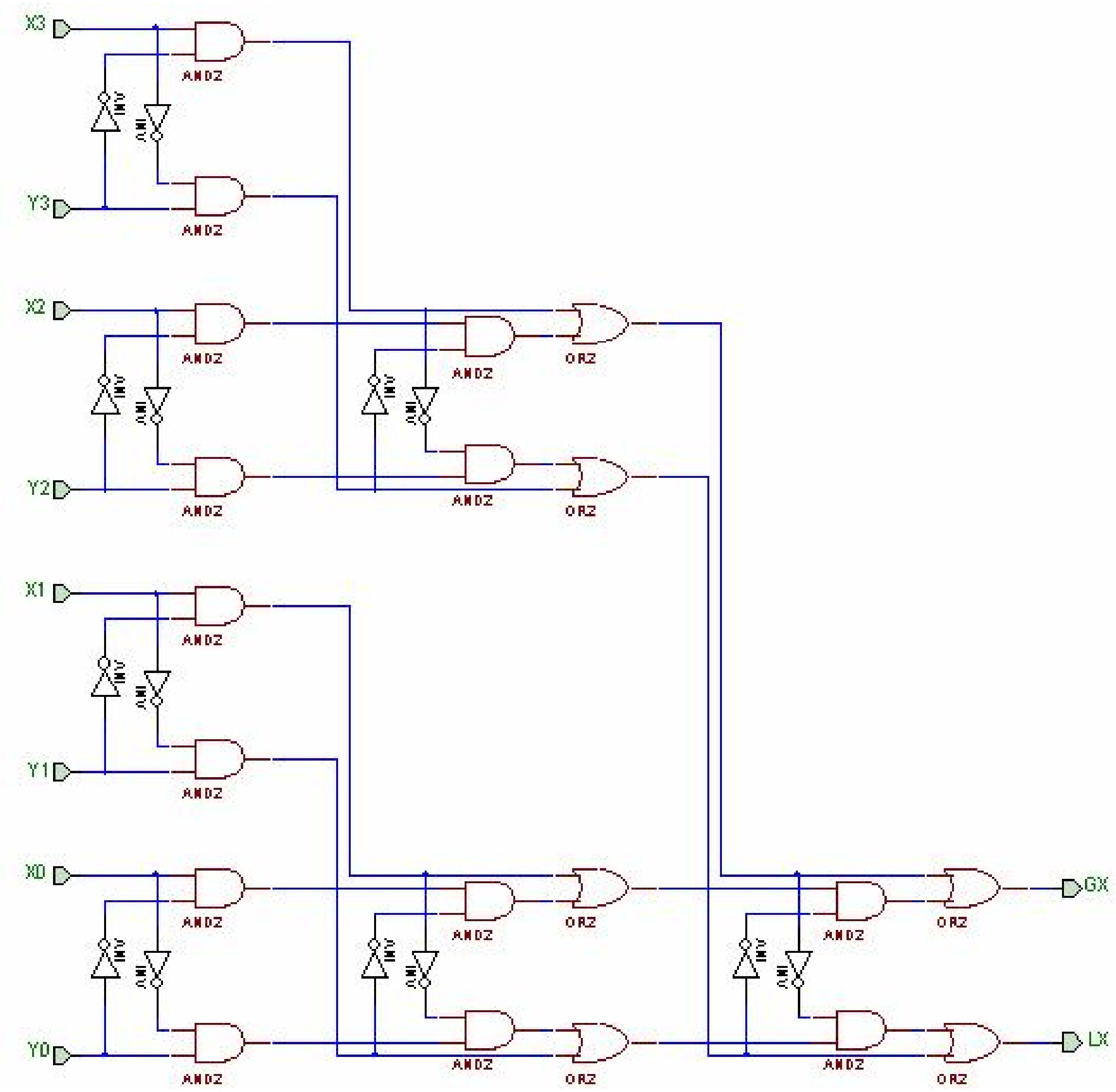


If every gate has a worst-case delay of 1 ns, what is the longest possible delay in your 4 bit circuit from the time the input values change to the time the output values finish changing? What is the worst-case delay for an *n* bit version of your circuit? Show how to build a faster *lookahead* version of your circuit, using only simple gates.

One of longest path from an input to an output is from input x_{n-1} through the inverter and the lower AND gate, through the inverter and upper AND gate in the next stage, through the inverter and lower AND gate in the next stage, and so forth. This path has 3n gates. So, with a 1 ns delay per gate, the longest possible delay is 12 ns and for an n bit comparator, the worst-case delay is 3n ns. We can build a lookahead version of the circuit we first need to derive the appropriate logic equations. Let $L_i = x_i'y_i$ and $G_i=x_iy_i'$. Then, for the 4 bit comparator, we can write $L = L_3+G_3'L_2 + G_3'G_2'L_1 + G_3'G_2'G_1'L_0$ and $G = G_3+L_3'G_2 + L_3'L_2'G_1 + L_3'L_2'L_1'G_0$. A circuit implementing these equations appears below. This can be converted to a circuit using simple gates by replacing each of the k input gates (for k>2) with a balanced tree of k-1 simple gates. This circuit has a worst-case delay of 7 ns, which is actually worse than the earlier circuit, but an for larger n it will be faster. The n bit version of the circuit has a worst-case delay of $3 + 2\log_2 n$ for any n that is a power of 2.



There is an alternative design for a lookahead comparator that requires fewer gates than the n bit version of the design shown above, which requires about n^2 simple gates. This one is based on a different way of writing the logic equations. Define L(i,j) to be true if the number formed from bits $x_{i+j-1}...x_i$ is less than the number formed from bits $y_{i+j-1}...y_i$. Define G(i,j) similarly. Then for an n bit comparator, the final outputs L = L(0,n) and G = G(0,n). Now if n is a power of 2, we can write L(0,n) = L(n/2,n/2) + G(n/2,n/2)'L(0,n/2) and G(0,n) = G(n/2,n/2) + L(n/2,n/2)'G(0,n/2). Similarly, L(0,n/2) = L(n/4,n/4) + G(n/4,n/4)'L(0,n/4) and L(0,n/2) = L(n/4,n/4) + G(n/4,n/4)'L(0,n/4) and in general L(i,j) = L(i+j/2,j/2) + G(i+j//2,j/2)'L(i,j/2) and G(i,j) = G(i+j/2,j/2) + L(i+j//2,j/2)'G(i,j/2). A 4 bit version of the circuit obtained using these equations is shown below. The n bit version has a worst-case delay of 2 + 3 $\log_2 n$ which is not quite as good as the first lookahead circuit, but this circuit has a gate count of 8n – 5, which is a huge improvement.


X3
Y3
X2
Y2
X1
Y1
X0
Y0
AND2
INV
OR2
GX
LX