1. Sum the following binary (base 2) numbers

$$\begin{array}{r} 10011_2 \\ \underline{+10110_2} \end{array} \qquad\qquad \begin{array}{r} 101101_2 \\ \underline{+110111_2} \end{array}$$

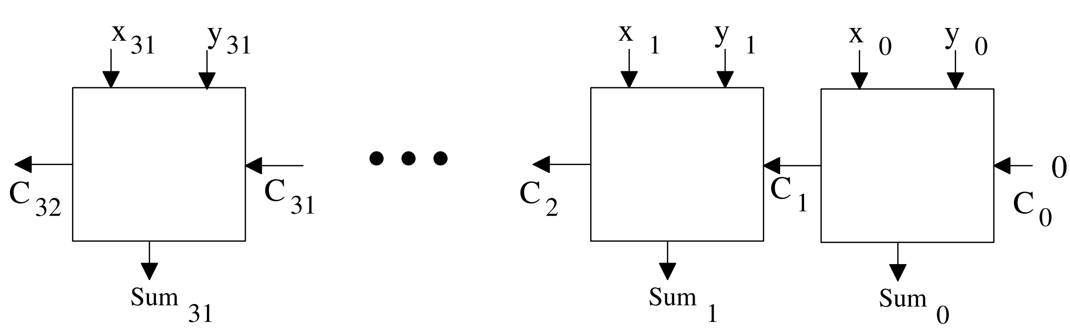
2. Complete the Full-Adder truth table for the sum (s_i) and carry-out (c_{i+1}) functions.

x_i	y_i	**carry-in** c_i	**sum** s_i	**carry-out** c_{i+1}
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

3. Use k-maps to minimize the sum (s_i) and carry-out (c_{i+1}) functions of the Full-Adder:

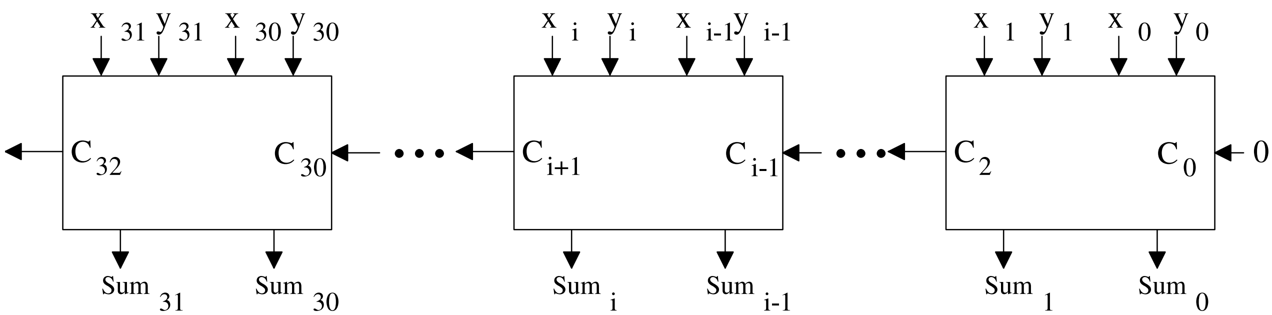
4. For the one-bit Full-Adder, how many gate delays are needed before the carry-out (c_{i+1}) wire is correct?

5. A 32-bit, ripple-adder is made up of a collection of single-bit Full-Adders connected together as shown below:



How many gate delays are needed before c_{32} is correct?

6. To speed up the calculation of the carry-out (C_{i+1}) signals, consider constructing a 32-bit adder using two-bit adders as shown in:



If c_{i+1} is calculated directly from the inputs as $c_{i+1} = x_i y_i + x_i x_{i-1} y_{i-1} + x_i x_{i-1} c_{i-1} + x_i y_{i-1} c_{i-1} + y_i x_{i-1} y_{i-1} + y_i x_{i-1} c_{i-1} + y_i y_{i-1} c_{i-1}$, then how many gate delays would be needed to calculate the c_{i+1} signal in a two-bit adder?

7. What would be the total number of gate delays in a 32-bit adder before the c_{32} signal is generated correctly if **two-bit adders** were used?

8. What would be the total number of gate delays in a 32-bit adder before the c_{32} signal is generated correctly if **three-bit adders** were used (10 three-bit adders and a 2-bit adder)?

9. What would be the total number of gate delays in a 32-bit adder before the c_{32} signal is generated correctly if **four-bit adders** were used?