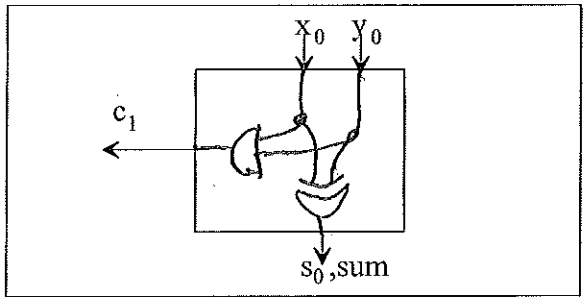
1. Sum the following binary (base 2) numbers

10011_2 X
$+10110_2$ Y

101101_2
$+110111_2$

2. Complete the Half-Adder truth table and circuit diagram.

x_0	y_0	sum s_0	carry-out c_1
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



3. 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
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

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

SUM

x_i \ $y_i c_i$	00	01	11	10
0	0	1	0	1
1	1	0	1	0

XOR pattern

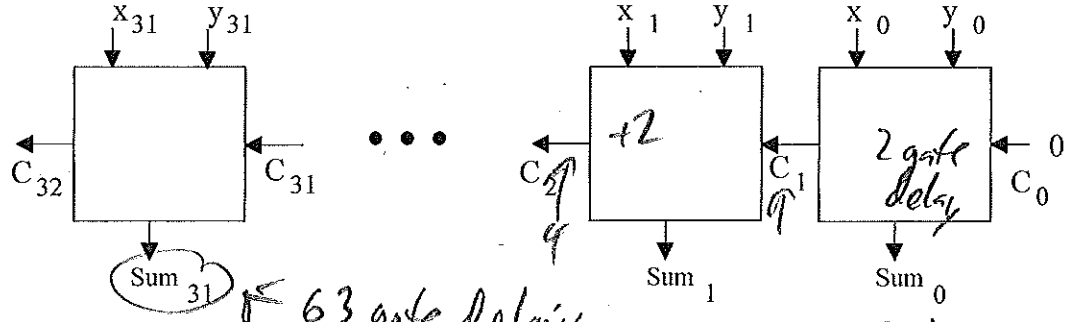
C_{i+1}:

x_i \ $y_i c_i$	00	01	11	10
0	0	0	1	0
1	0	1	1	1

$C_{i+1} = x_i c_i + y_i c_i + x_i y_i$

5. For the one-bit Full-Adder, how many gate delays are needed before the carry-out (c_{i+1}) wire is correct? 2

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

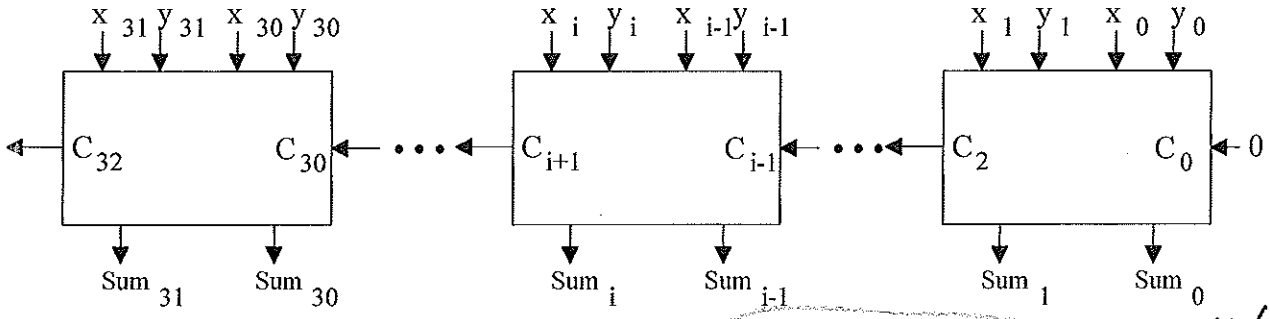


63 gate delays

How many gate delays are needed before c_{32} is correct? 64 gate delays

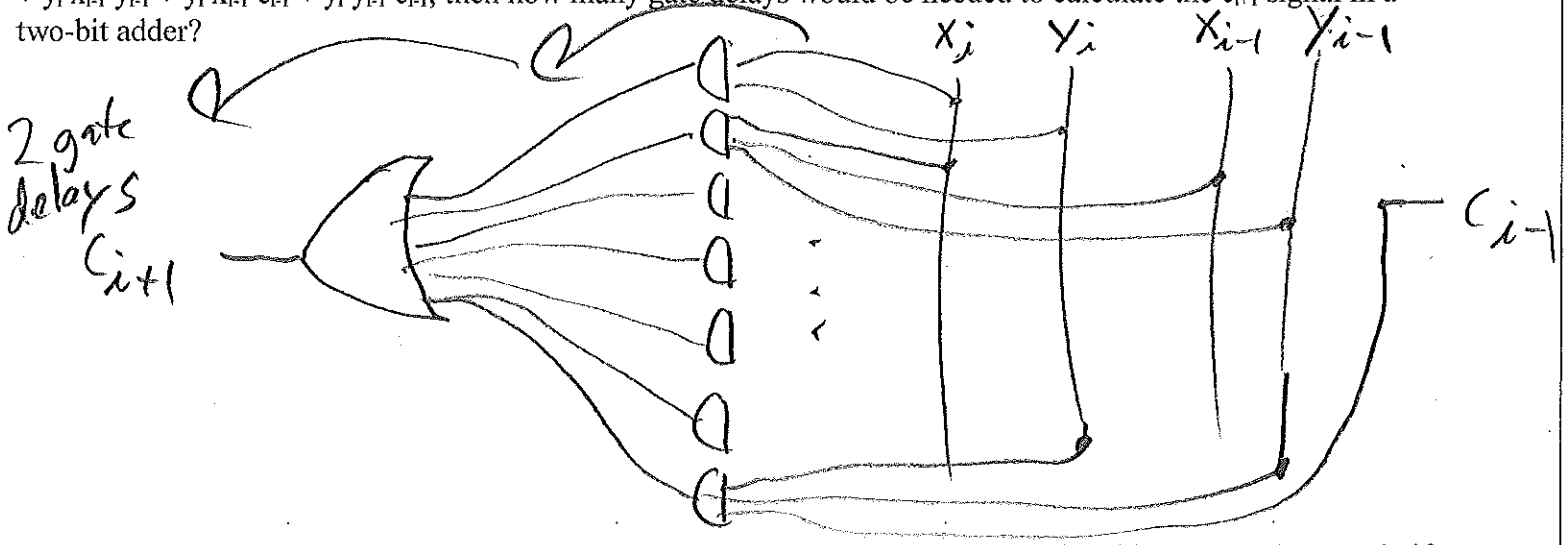
32 x 2 =

7. 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?

x_i (3-carry in C_i)

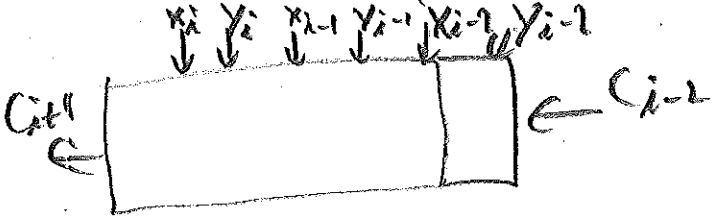


8. What would be the total number of gate delays in a 32-bit adder before the c_{32} signal is generated correctly if **two-bit adders** were used?

$16 \times 2 = 32$

16 2-bit adders needed to get to 32-bits. Each add 2 gate delays

9. 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)?



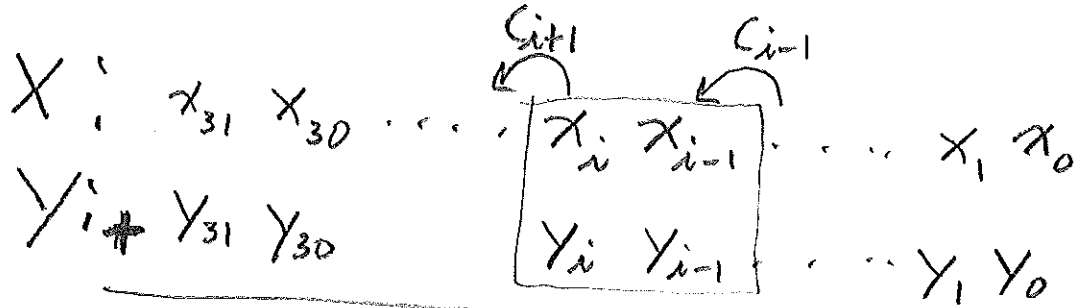
$10 \times 3 + 2 = 32$

10. 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?

$8 \times 3 = 24$

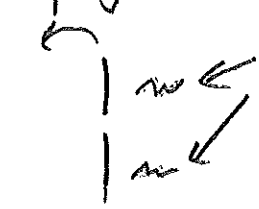
3 gate delays each

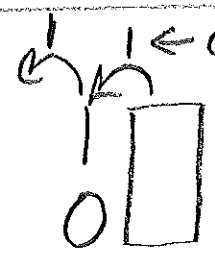
7. To understand the two-bit adder, keep in mind the addition being performed. We are considering 2-bits of a 32-bit addition to determine the carry-out (c_{i+1}) value of two-bits:



All the ways to get c_{i+1} to be a 1:

$x_i\ y_i \equiv$

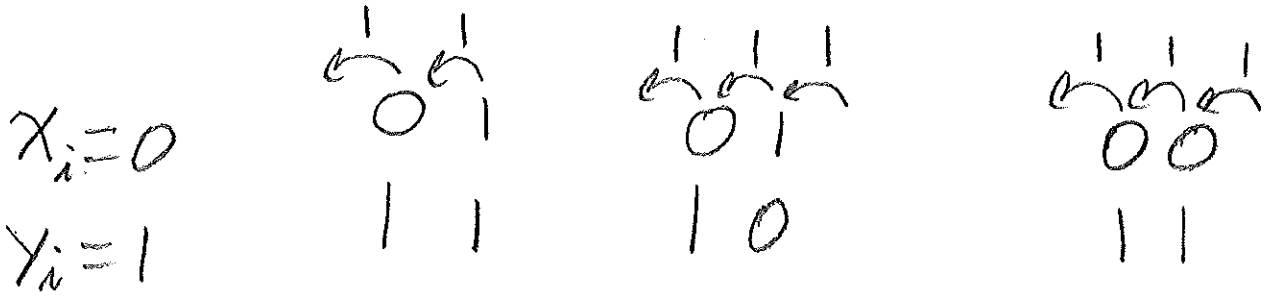




Three ways to get carry into i^{th} bit to be a 1:

$x_i\ x_{i-1}\ y_{i-1}$ $x_i\ x_{i-1}\ c_{i-1}$ $x_i\ y_i\ c_{i-1}$

Similarly, $x_i = 0$ and $y_i = 1$ with a carry in to i^{th} bit of a 1. Three ways to get the carry into i^{th} bit to be a 1:



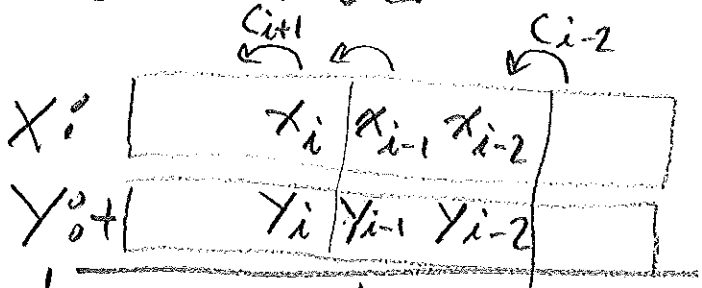
$$y_i x_{i-1} y_{i-1} \quad y_i x_{i-1} c_{i-1} \quad y_i y_{i-1} c_{i-1}$$

In summary the SoP formula for carry out of i^{th} bit, c_{i+1} is

$$c_{i+1} = x_i y_i + x_i x_{i-1} y_{i-1} + x_i x_{i-1} c_{i-1} + x_i y_{i-1} c_{i-1} + y_i x_{i-1} y_{i-1} + y_i x_{i-1} c_{i-1} + y_i y_{i-1} c_{i-1}$$

2 gate delays need to do all ANDs in parallel, then the 7-input OR.

(9) 3-bit adder



$$\begin{array}{r} 1xx \\ +\ 1xx \end{array} \qquad \begin{array}{r} 11x \\ 01y \end{array} \qquad \begin{array}{r} 1 \\ 100 \\ 011 \end{array}$$

$$c_{i+1} = x_i y_i + x_i x_{i-1} y_{i-1} + \cdots + x_i y_{i-1} y_{i-2} c_{i-2}$$

$$+ y_i (x_{i-1} y_{i-1} + \cdots + y_{i-1} y_{i-2} c_{i-2})$$

7-term SOP to get carry out of 1 from least-significant 2-bits

$$c_{i+1} = x_i y_i + x_i x_{i-1} y_{i-1} + \cdots + y_i y_{i-1} y_{i-2} c_{i-2}$$

15 product terms in SOP

(9) (3-bit add continued)

circuit for carry out, c_{i+1}

(NOTE: AND + OR gates have a 9-input limit. If more inputs needed, then you need to build one using several simpler gates.)