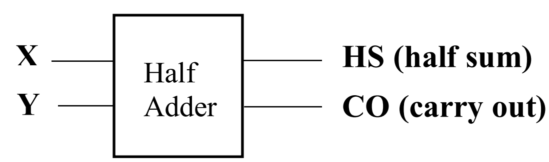
Binary Adders and Subtractors

<u>Outline:</u>

- Half Adder
- Full Adder
- Ripple Carry Adder
- Carry-Lookahead (CLA) Adder
- 74x283 4-bit Adder
- 16-bit Group-ripple Adder
- Subtractor

<u>Half Adder (HA):</u> Adds 2 bits. Has 2 inputs and 2 outputs.

X	0	0	1	1
+ Y	+ 0	+ 1	+ 0	+ 1
CO HS	0 0	0 1	0 1	1 0

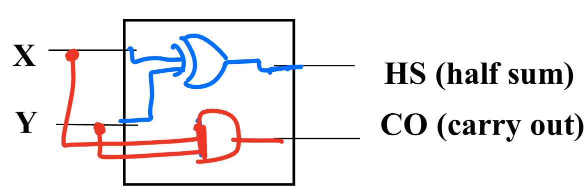


Constructing the truth table and finding Boolean expressions for HS and CO:

X	Y	HS	CO
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$HS = X \oplus Y$

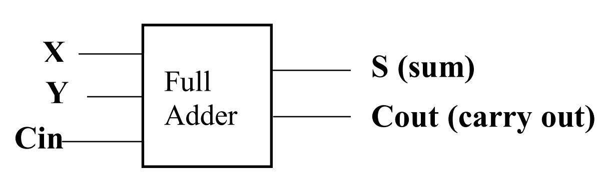
$CO = X \cdot Y$



<u>Full Adder (FA):</u> Adds 3 bits. Has 3 inputs and 2 outputs.

Cin (carry in)
+ X
+ Y

Cout S



Constructing the truth table for FA and finding Boolean expressions for S and Cout:

Cin	X	Y	S	Cout
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

Cin \ XY	00	01	11	10
0		1		1
1	1		1	

$S = X \oplus Y \oplus C_{in}$

Cin \ XY	00	01	11	10
0			1	
1		1	1	1

$C_{out} = X \cdot Y + C_{in} \cdot X + C_{in} \cdot Y$

$C_{in} \cdot Y$ $X \cdot Y$ $C_{in} \cdot X$

$$C_{out} = X \cdot Y + C_{in}(X+Y) = X \cdot Y + C_{in}\left[(X+Y)\cdot(X+X')\right]$$

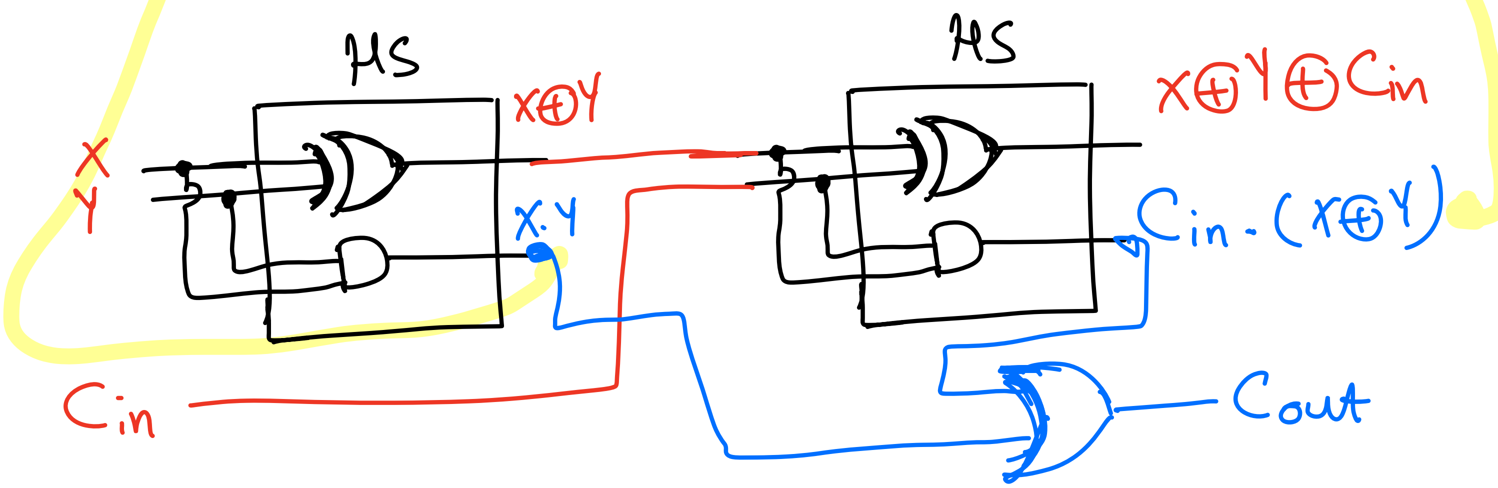
$$= X \cdot Y + C_{in}\left[X + X' \cdot Y\right] = X \cdot Y + C_{in} \cdot X + C_{in} \cdot X' \cdot Y$$

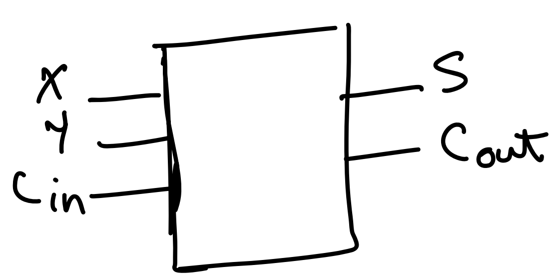
$$= X \cdot Y + C_{in} \cdot X\left[Y + Y'\right] + C_{in} \cdot X' \cdot Y = \underline{X \cdot Y + C_{in} \cdot X \cdot Y} + C_{in} \cdot X \cdot Y' + C_{in} \cdot X' \cdot Y$$

$$= X \cdot Y\left[C_{in} + 1\right] + C_{in}\left[X \cdot Y' + X' \cdot Y\right]$$

$$= X \cdot Y + C_{in}(X \oplus Y)$$

Implementing a Full Adder with Half Adders:

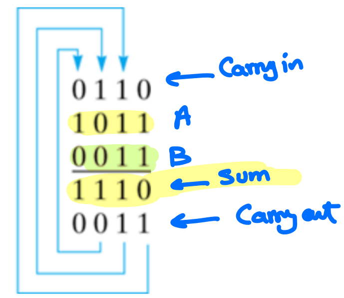




Ripple Carry Adder: cascading Full Adders, with the carry output from one Full Adder connected to the carry input of the next Full Adder.

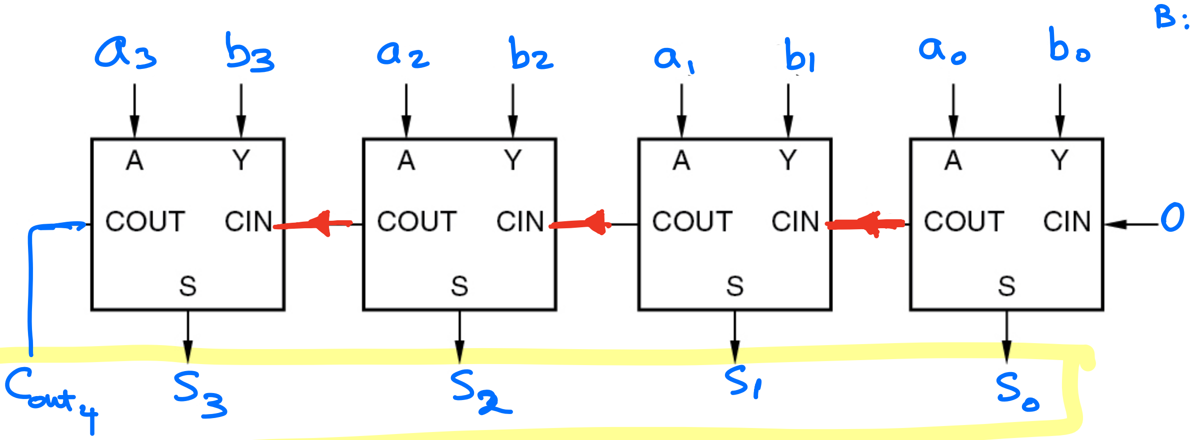
A 4-bit Ripple Carry Adder:

$$\begin{array}{r} a_3\ a_2\ a_1\ a_0 \\ +\ \ b_3\ b_2\ b_1\ b_0 \\ \hline c_4\ s_3\ s_2\ s_1\ s_0 \end{array}$$



The input carry in the least significant position is 0. The output carry in each position is the input carry of the next-higher-order position.

A: $a_3 a_2 a_1 a_0$
B: $b_3 b_2 b_1 b_0$

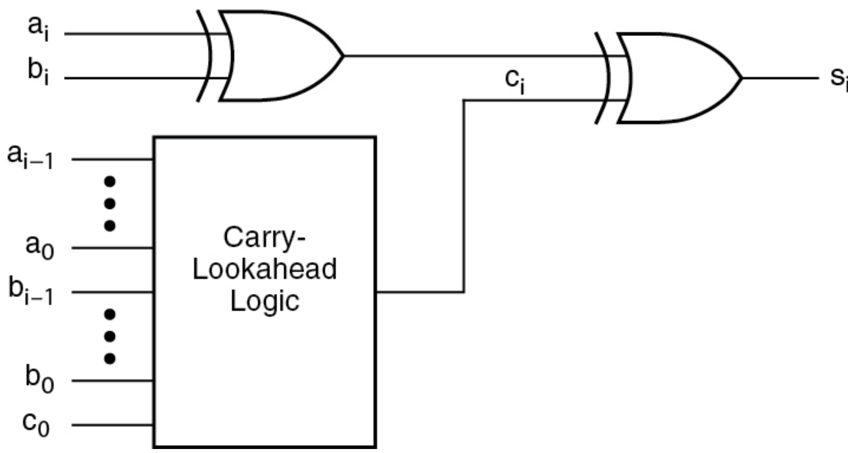


A ripple adder is slow. In the worst case, a carry must propagate from the least significant Full Adder to the most significant one:

$$\begin{array}{r} 1\,1\,1\,1 \\ +\,0\,0\,0\,1 \\ \hline \end{array}$$

R

Carry-Lookahead Adder: Calculates the carryout for each stage ahead of addition to speed up the computation, but it requires more complex circuits.



Some Definitions:

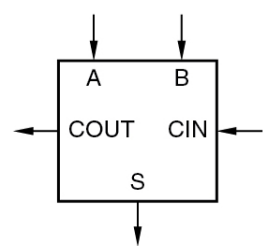
carry generate (g_i):

For a particular combination of inputs a_i and b_i, the adder stage i is said to *generate* a carry if it produces a $c_{out}=1$ (that is $c_{i+1}=1$), independent of what lower-order input bits ($a_0,\ldots,a_{i-1}$ and, $b_0,\ldots b_{i-1}$, c_0) are.

carry propagate (p_i):

For a particular combination of inputs a_i and b_i, the adder stage i is said to *propagate* a carry if it produces a $c_{out}=1$ (that is $c_{i+1}=1$), when $c_{in}=1$ (that is $c_i=1$).

For adder stage *i*:



a_i	b_i	c_i	s_i	c_{i+1}	
0	0	0	0	0	
0	0	1	1	0	
0	1	0	1	0	
0	1	1	0	1	propagate
1	0	0	1	0	
1	0	1	0	1	propagate
1	1	0	0	1	generate
1	1	1	1	1	generate / propagate

$g_i = a_i \cdot b_i$

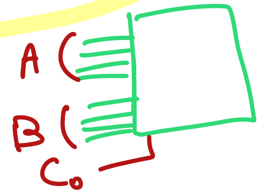
$p_i = a_i \oplus b_i$

$\boxed{c_{i+1} = g_i + c_i \cdot p_i}$

$= a_i \cdot b_i + c_i \cdot (a_i \oplus b_i)$

(when consider the last term as generate)

If last term was considered as propagate, then $P_i = a_i + b_i$

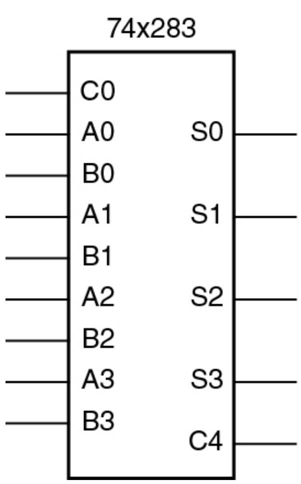


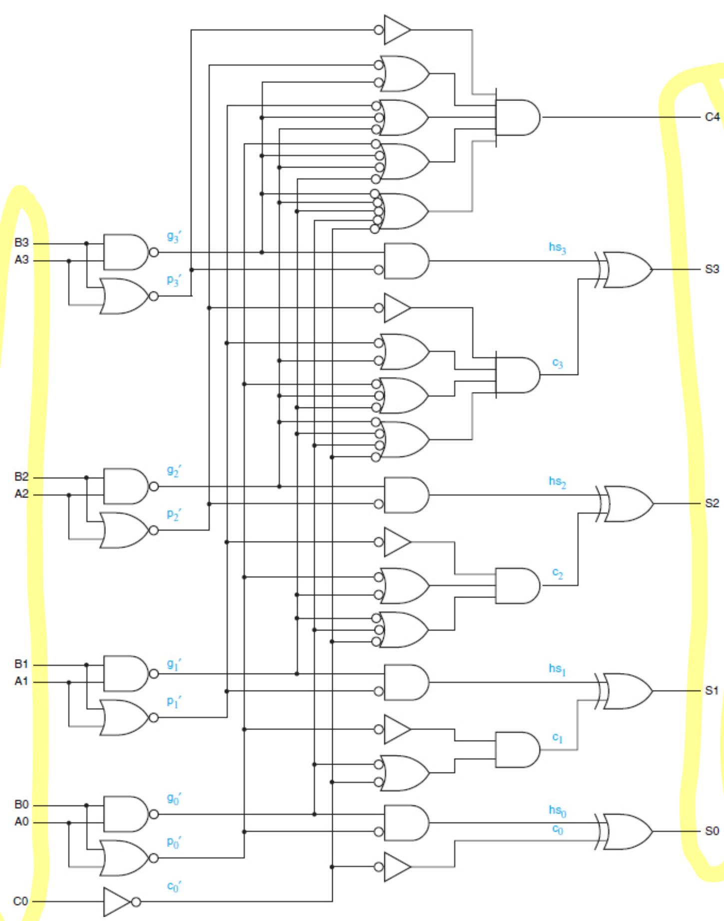
Carry equations in a 4-bit CLA adder:

$$
\begin{aligned}
c_1 &= g_0 + C_0 \cdot P_0 = a_0 \cdot b_0 + C_0 \cdot (a_0 \oplus b_0) \\
c_2 &= g_1 + C_1 \cdot P_1 \\
&= a_1 \cdot b_1 + [a_0 \cdot b_0 + C_0 (a_0 \oplus b_0)] \cdot (a_1 \oplus b_1) \\
&= \\
c_3 &= g_2 + p_2 \cdot c_2 \\
&= g_2 + p_2 \cdot (g_1 + p_1 \cdot g_0 + p_1 \cdot p_0 \cdot c_0) \\
&= g_2 + p_2 \cdot g_1 + p_2 \cdot p_1 \cdot g_0 + p_2 \cdot p_1 \cdot p_0 \cdot c_0 \\
c_4 &= g_3 + p_3 \cdot c_3 \\
&= g_3 + p_3 \cdot (g_2 + p_2 \cdot g_1 + p_2 \cdot p_1 \cdot g_0 + p_2 \cdot p_1 \cdot p_0 \cdot c_0) \\
&= g_3 + p_3 \cdot g_2 + p_3 \cdot p_2 \cdot g_1 + p_3 \cdot p_2 \cdot p_1 \cdot g_0 + p_3 \cdot p_2 \cdot p_1 \cdot p_0 \cdot c_0
\end{aligned}
$$

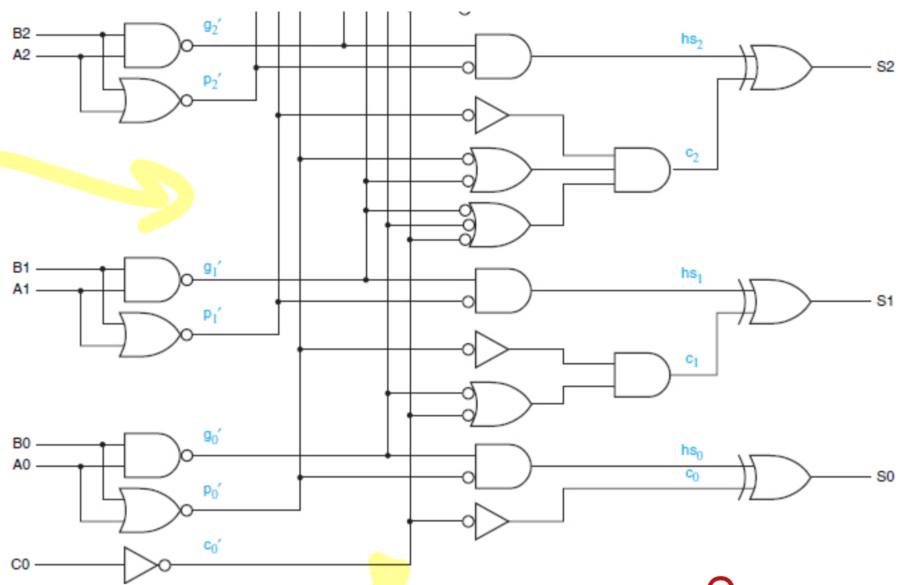
<u>74x283 4-bit Adder:</u>

- This is a 4-bit CLA adder.
- Generates active-low version for "g_i" and "p_i"
- Uses the "OR" version for implementing "p_i"





A more detailed look:



$p_i = a_i + b_i$

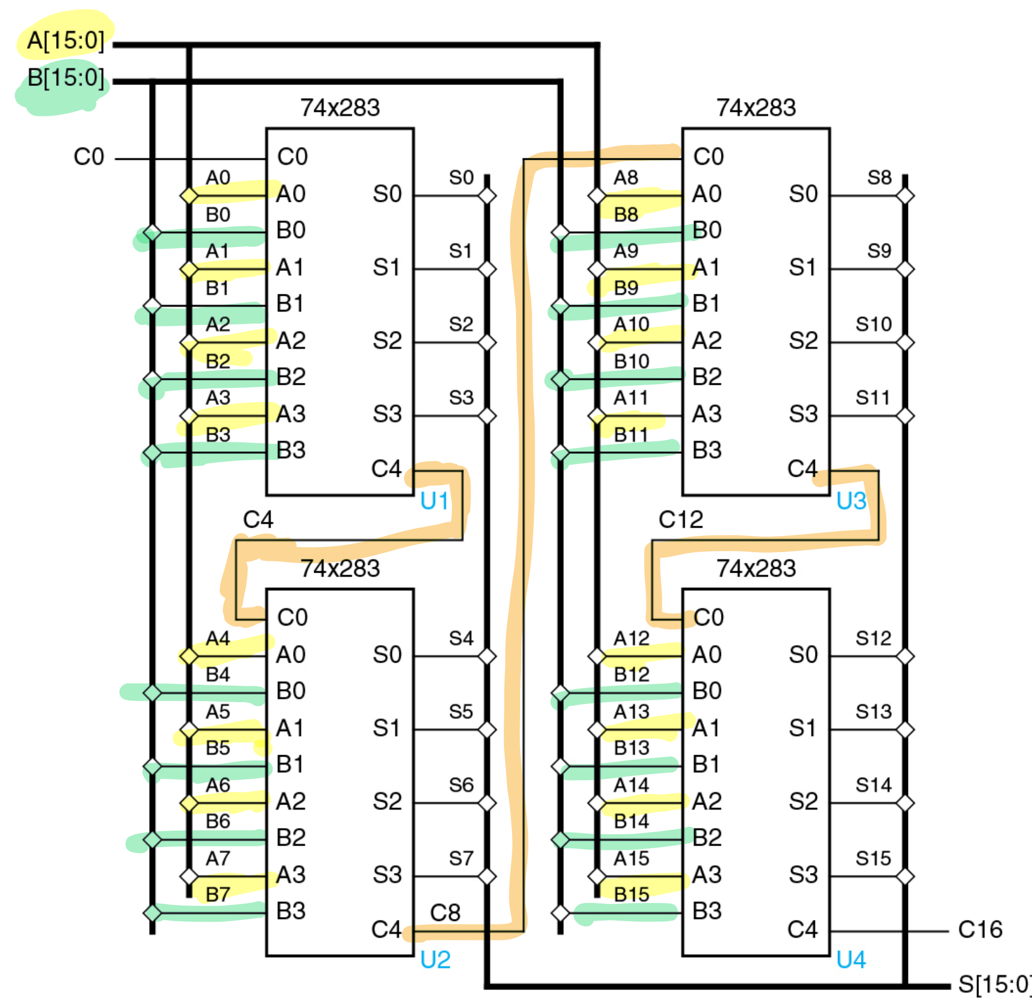
$g_i = a_i \cdot b_i$

Carry equations in a 74x283 adder:

$$
\begin{aligned}
c_1 &= p_0 \cdot (g_0 + c_0) \\
c_2 &= p_1 \cdot (g_1 + c_1) \\
&= p_1 \cdot (g_1 + p_0 \cdot (g_0 + c_0)) \\
&= p_1 \cdot (g_1 + p_0) \cdot (g_1 + g_0 + c_0) \\
c_3 &= p_2 \cdot (g_2 + c_2) \\
&= p_2 \cdot (g_2 + p_1 \cdot (g_1 + p_0) \cdot (g_1 + g_0 + c_0)) \\
&= p_2 \cdot (g_2 + p_1) \cdot (g_2 + g_1 + p_0) \cdot (g_2 + g_1 + g_0 + c_0) \\
c_4 &= p_3 \cdot (g_3 + c_3) \\
&= p_3 \cdot (g_3 + p_2 \cdot (g_2 + p_1) \cdot (g_2 + g_1 + p_0) \cdot (g_2 + g_1 + g_0 + c_0)) \\
&= p_3 \cdot (g_3 + p_2) \cdot (g_3 + g_2 + p_1) \cdot (g_3 + g_2 + g_1 + p_0) \cdot (g_3 + g_2 + g_1 + g_0 + c_0)
\end{aligned}
$$

16-bit Group-Ripple Adder:

Building 16-bit Adder by cascading 4 4-bit CLA adders:

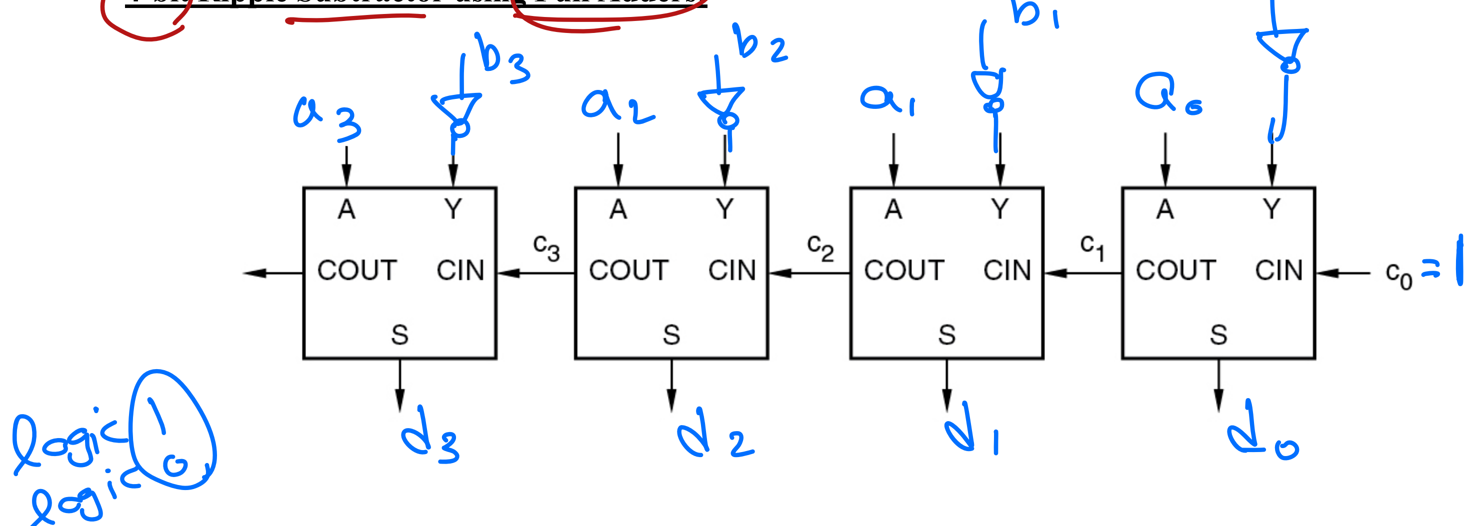


Two's-Complement Subtraction:

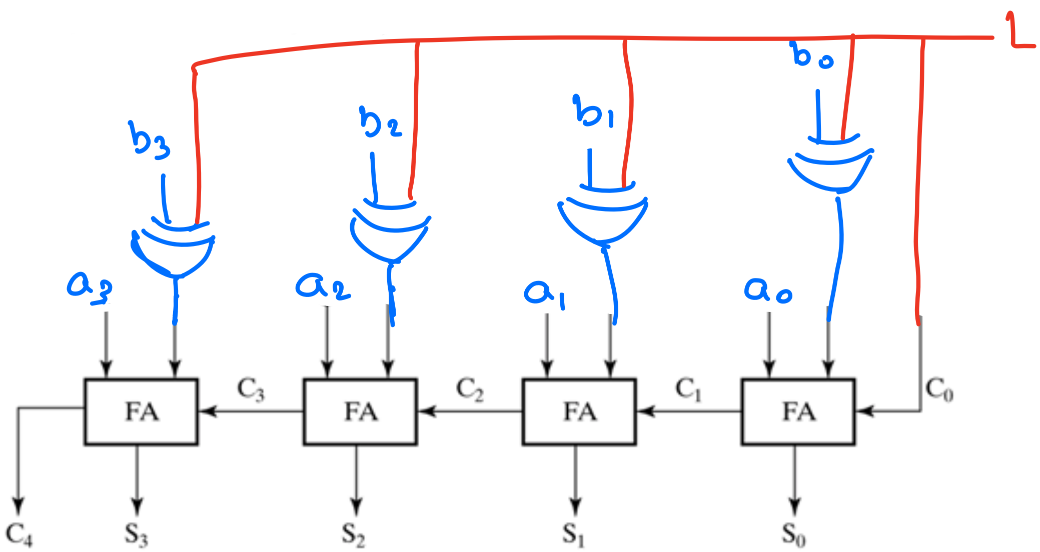
$$A = a_3 a_2 a_1 a_0, \quad B = b_3 b_2 b_1 b_0$$

$$\begin{array}{r} a_3\ a_2\ a_1\ a_0 \\ -\ b_3\ b_2\ b_1\ b_0 \\ \hline \end{array} \equiv \begin{array}{r} a_3\ a_2\ a_1\ a_0 \\ +\ b_3'\ b_2'\ b_1'\ b_0' \\ +\ 1 \\ \hline d_3\ d_2\ d_1\ d_0 \end{array}$$

4-bit Ripple Subtractor using Full Adders:



4-bit Ripple Adder/Subtractor:



If $L = 0$

$$\left.\begin{array}{l} b_0 \oplus 0 = b_0 \\ b_1 \oplus 0 = b_1 \\ b_2 \oplus 0 = b_2 \\ b_3 \oplus 0 = b_3 \end{array}\right) \equiv \text{addition}$$

If $L = 1$

$$\left.\begin{array}{l} b_0 \oplus 1 = b_0' \\ b_1 \oplus 1 = b_1' \\ b_2 \oplus 1 = b_2' \\ b_3 \oplus 1 = b_3' \end{array} \quad C_0 = 1 \right) \equiv \text{2's-complement Subtraction}$$

$$X \oplus Y = X.Y' + X'.Y$$

$$X \oplus 1 = X.0 + X'.1$$

$$= 0 + X'$$

$$= X'$$