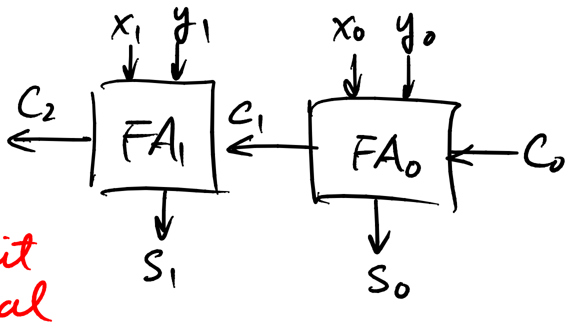
Carry-ripple & Carry-lookahead adder

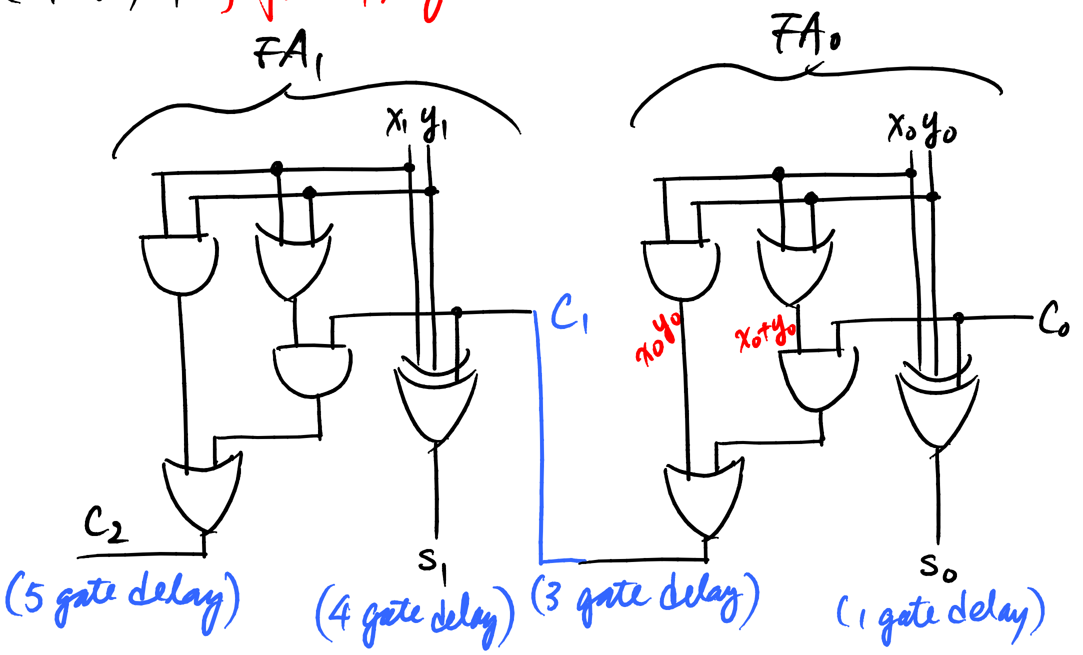
$S_0 = x_0 \oplus y_0 \oplus C_0$

$C_1 = x_0 y_0 + x_0 C_0 + y_0 C_0$

$\quad = x_0 y_0 + (x_0 + y_0) C_0$



$$\left.\begin{aligned} S_1 &= x_1 \oplus y_1 \oplus C_1 \\ C_2 &= x_1 y_1 + (x_1 + y_1) C_1 \end{aligned}\right\} \text{need to wait for } C_1 \text{ signal}$$



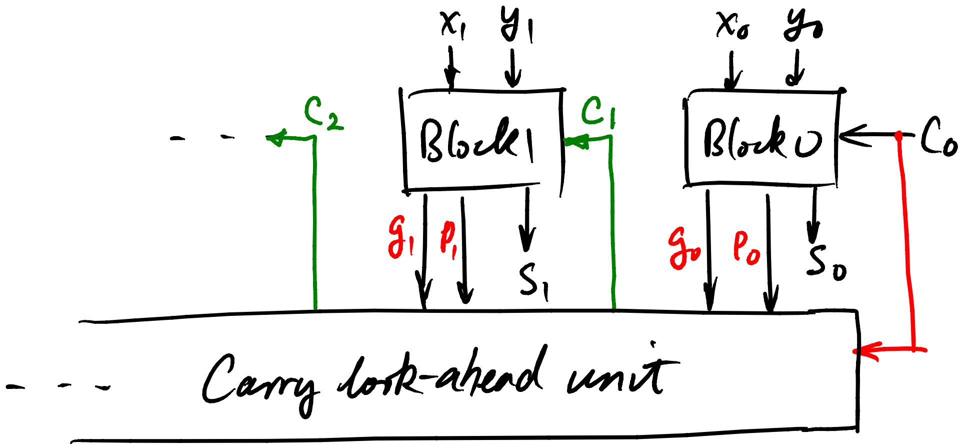
For a 16-bit adder: C_{16} ($16 \times 2 + 1 = 33$ gate delay)

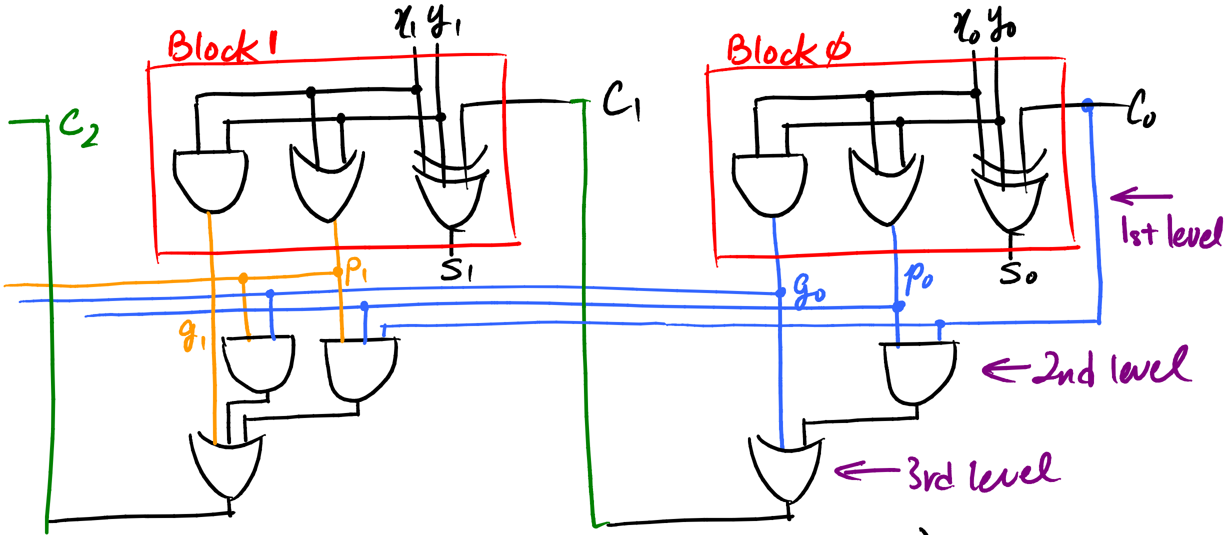
Carry look-ahead adder (Fast adder)

$$\left.\begin{aligned} S_0 &= x_0 \oplus y_0 \oplus C_0 \\ C_1 &= x_0 y_0 + x_0 C_0 + y_0 C_0 \\ &= x_0 y_0 + (x_0 + y_0) C_0 \xrightarrow{\quad} g_0 + P_0 C_0 \end{aligned}\right\} FA_0$$

$S_1 = x_1 \oplus y_1 \oplus C_1$

$C_2 = x_1 y_1 + (x_1 + y_1) C_1 = \overset{g_1}{\overset{\parallel}{x_1 y_1}} + \overbrace{(x_1 + y_1)}^{P_1}(\overbrace{x_0 y_0}^{g_0} + \overbrace{(x_0 + y_0)}^{P_0} C_0) = g_1 + P_1 g_0 + P_1 P_0 C_0$





⇒ for 16 bit fast adder C_{16} (3 gate-level delay)

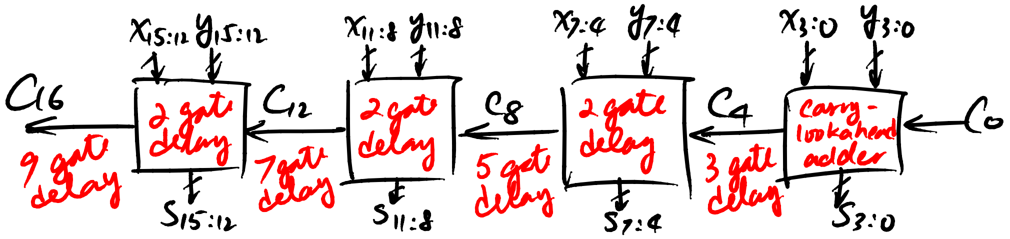
Pro's: fewer gate level delays

Con's: $\underbrace{\text{large number of inputs}}_{\text{fan-in}}$ on gates.

→ usually fan-in of 4 is acceptable.

What to do about a 16-bit adder?

Option 1: Comb. of look-ahead and carry ripple

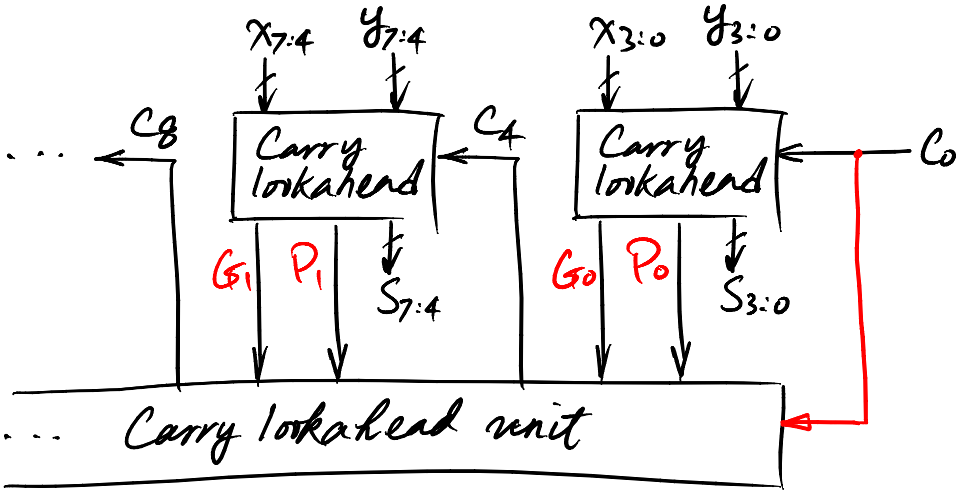


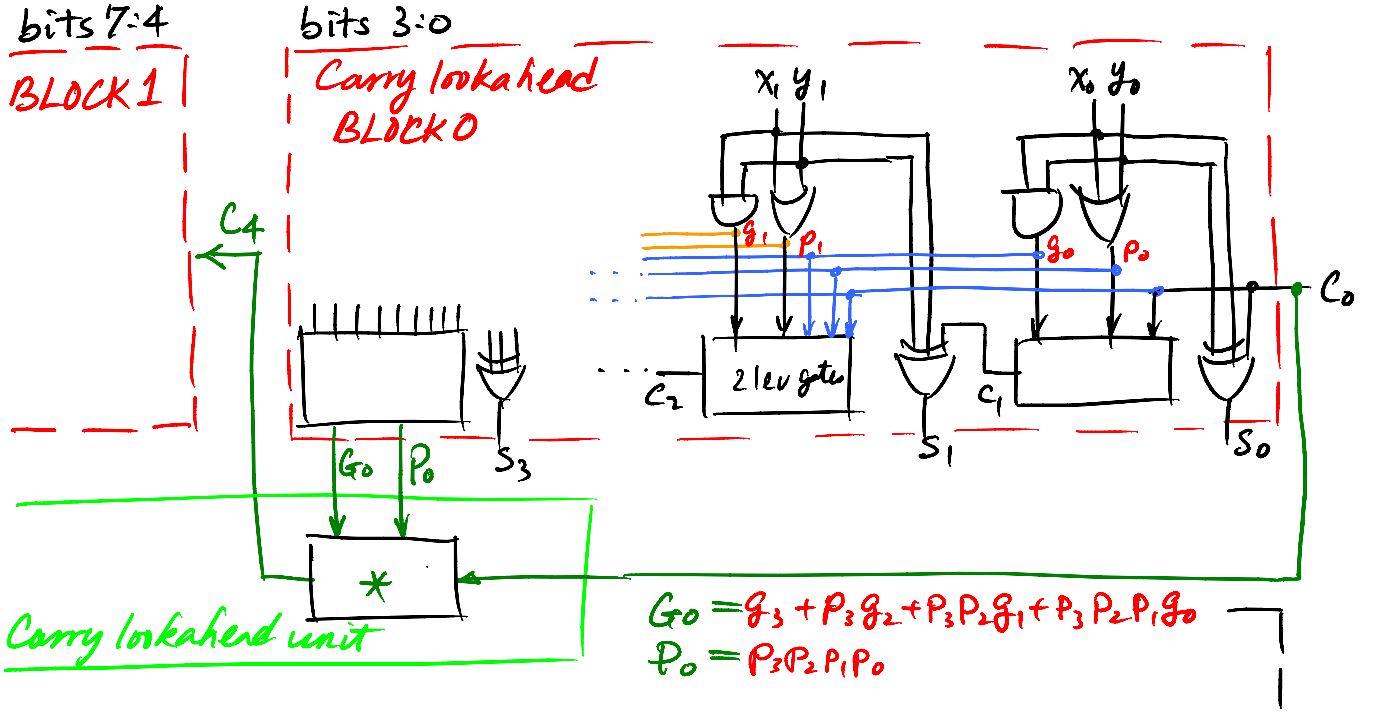
lookahead adder

$$C_4 = g_3 + P_3 C_3 = \underbrace{g_3 + P_3 g_2 + P_3 P_2 g_1 + P_3 P_2 P_1 g_0 + P_3 P_2 P_1 P_0 C_0}$$

→ 1 gate level delay to generate g_i, P_i
→ 1 gate level delay to generate all product terms
→ 1 gate level delay to do the sum terms

Option 2: Hierarchical look-ahead





$G_0 = g_3 + P_3 g_2 + P_3 P_2 g_1 + P_3 P_2 P_1 g_0$
$P_0 = P_3 P_2 P_1 P_0$

$$* \; C_4 = \underbrace{G_0 + \underbrace{P_0 C_0}_{\text{1 gate}}}_{\text{1 gate}}$$

$$C_8 = G_1 + P_1 C_4 = G_1 + P_1(G_0 + P_0 C_0) = G_1 + P_1 G_0 + P_1 P_0 C_0$$

where $G_1 = g_7 + p_7 g_6 + p_7 p_6 g_5 + p_7 p_6 p_5 g_4$ ← +4 to index

$P_1 = p_7 p_6 p_5 p_4$

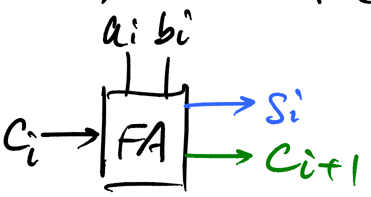
gate-level delays = 5 for C_{16}.

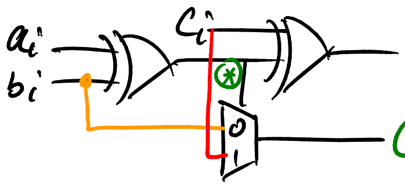
quick contrasts: (C_{16}) gate-level delay

carry-ripple :	33	
lookahead :	3	(large fan-in)
(opt 1) ripple-lookahead :	9	(fan-in ≤ 5)
(opt 2) lookahead-lookahead :	5	(fan-in ≤ 5)

	C_i	b_i	a_i	C_{i+1}
	0	0	0	0
(*)	0	0	1	0
	0	1	0	0
	0	1	1	1
	1	0	0	0
(*)	1	0	1	1
	1	1	0	1
	1	1	1	1

→ quick recall FA (Full Adder)

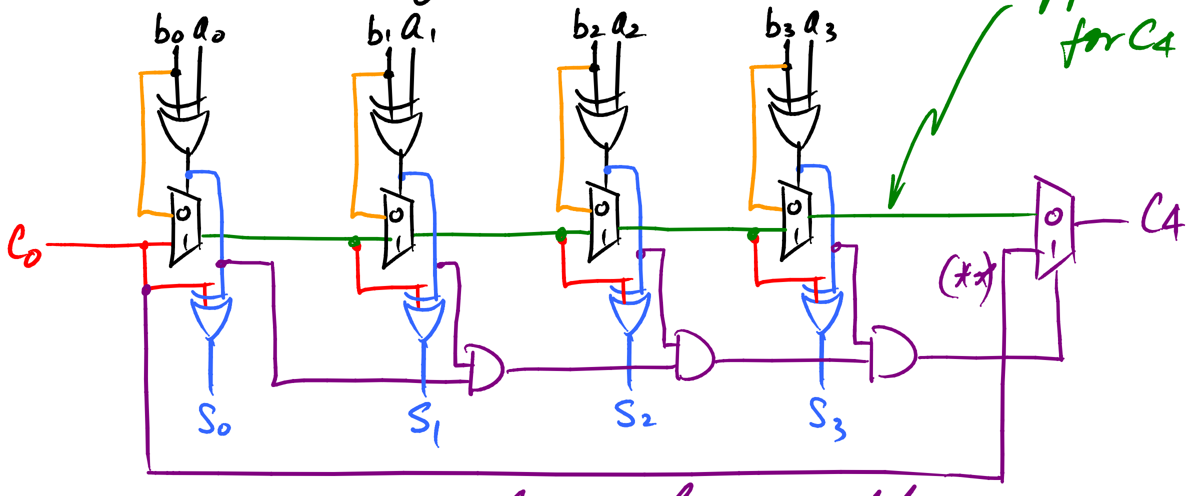




$S_i = a_i \oplus b_i \oplus C_i$

$C_{i+1} = a_i b_i + (a_i + b_i) C_i$

4-bit adder using the above FA.

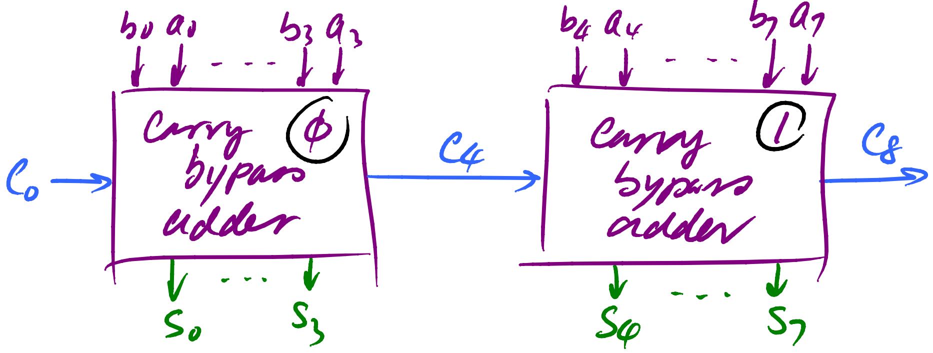


"ripple" carry path for C_4

carry-bypass adder

if a_i, b_i are different, (1,0) or (0,1), $C_4 = C_0$ (**)

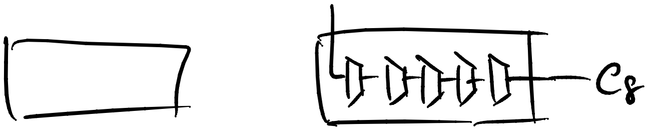
for a larger adder (8 bits)



Q: What is the worst-case delay through the 8-bit adder?

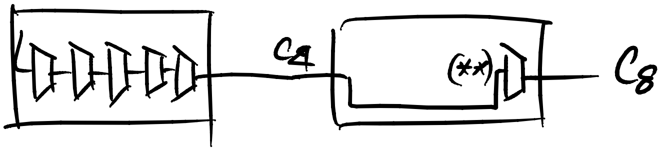
A: look at the carry-bypass adder ①

case 1: when any one $a_i b_i$ pair have the same value.



C_8 is independent of C_4

case 2: when all pairs $a_i b_i$ are different, i.e. (1,0) or (0,1).



$C_8 = C_4$

bypass path (**)