Carry Lookahead Adder (2A)

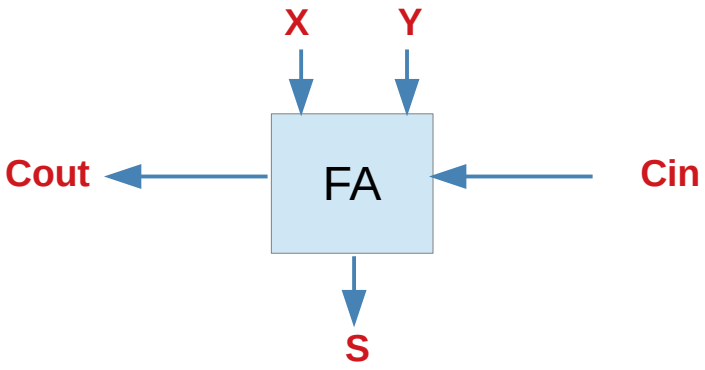
-
-

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using OpenOffice and Octave.

Carry Kill, Propagate, Generate conditions (1)

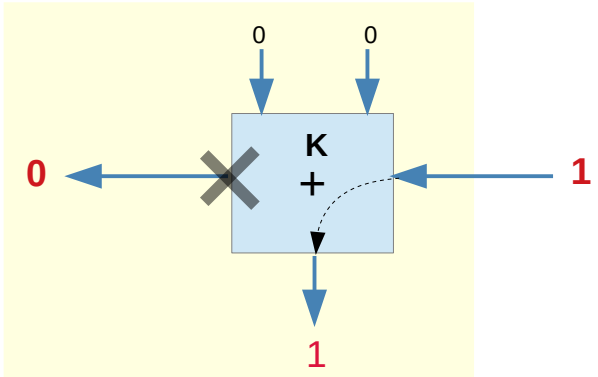
X	Y		
0	0	K	Kill (=$\overline{P}\overline{G}$)
0	1	P	Propagate
1	0	P	Propagate
1	1	G	Generate



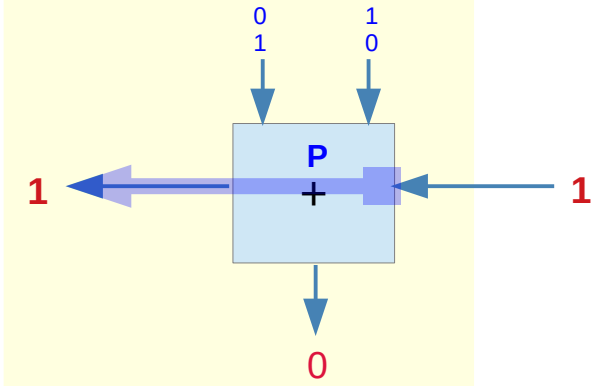
https::/electronics.stackexchange.com/questions/21251/critical-path-for-carry-skip-adder

Carry Kill, Propagate, Generate conditions (2)

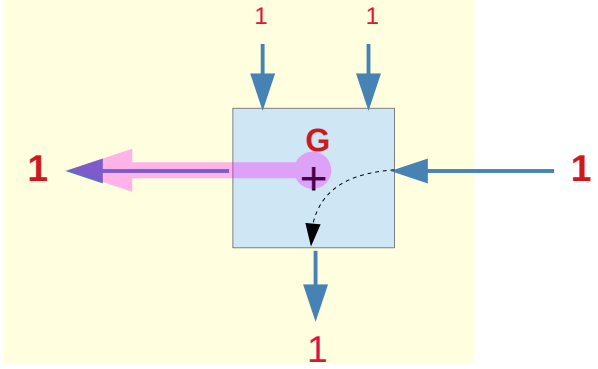
killed carry



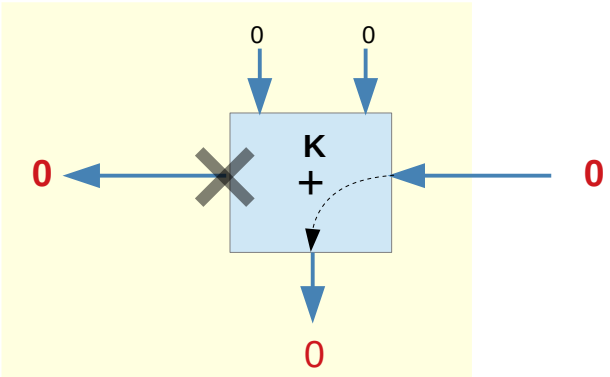
propagated carry



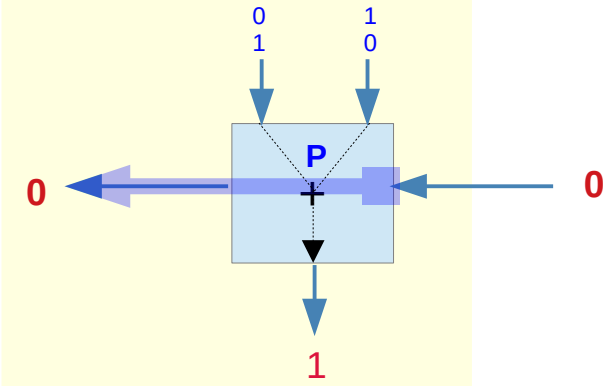
generated carry



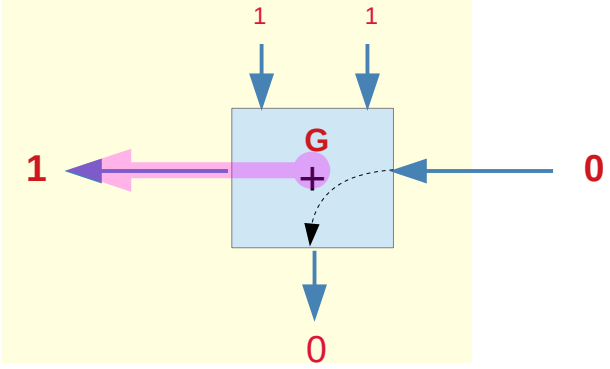
killed carry



propagated carry

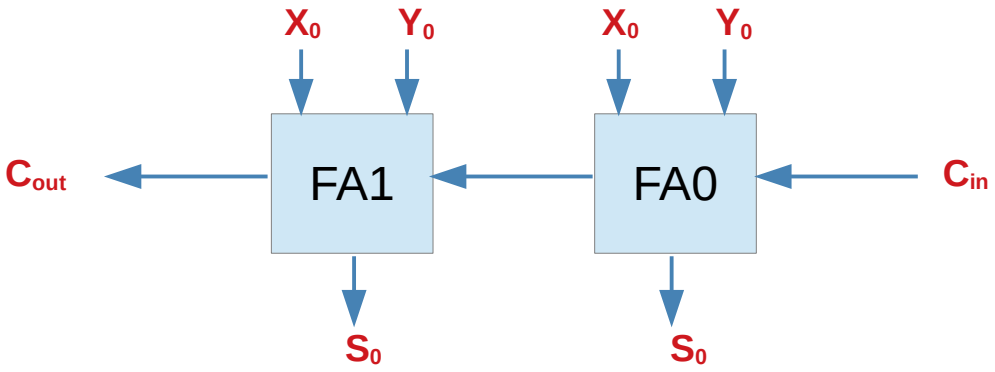


generated carry



K, **P**, and **G** conditions in a 2-bit adder (1)

X	Y		
0	0	K	Kill (=$\overline{P}\overline{G}$)
0	1	P	Propagate
1	0	P	Propagate
1	1	G	Generate



Unless the two FA's are in propagate mode, the transition of Cin does not affect the transition of Cout

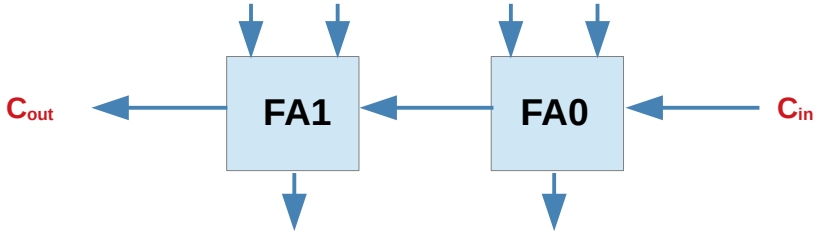
Critical path – all FA's in propagate mode

Broken paths for any FA in other mode
- kill mode, generate mode

https::/electronics.stackexchange.com/questions/21251/critical-path-for-carry-skip-adder

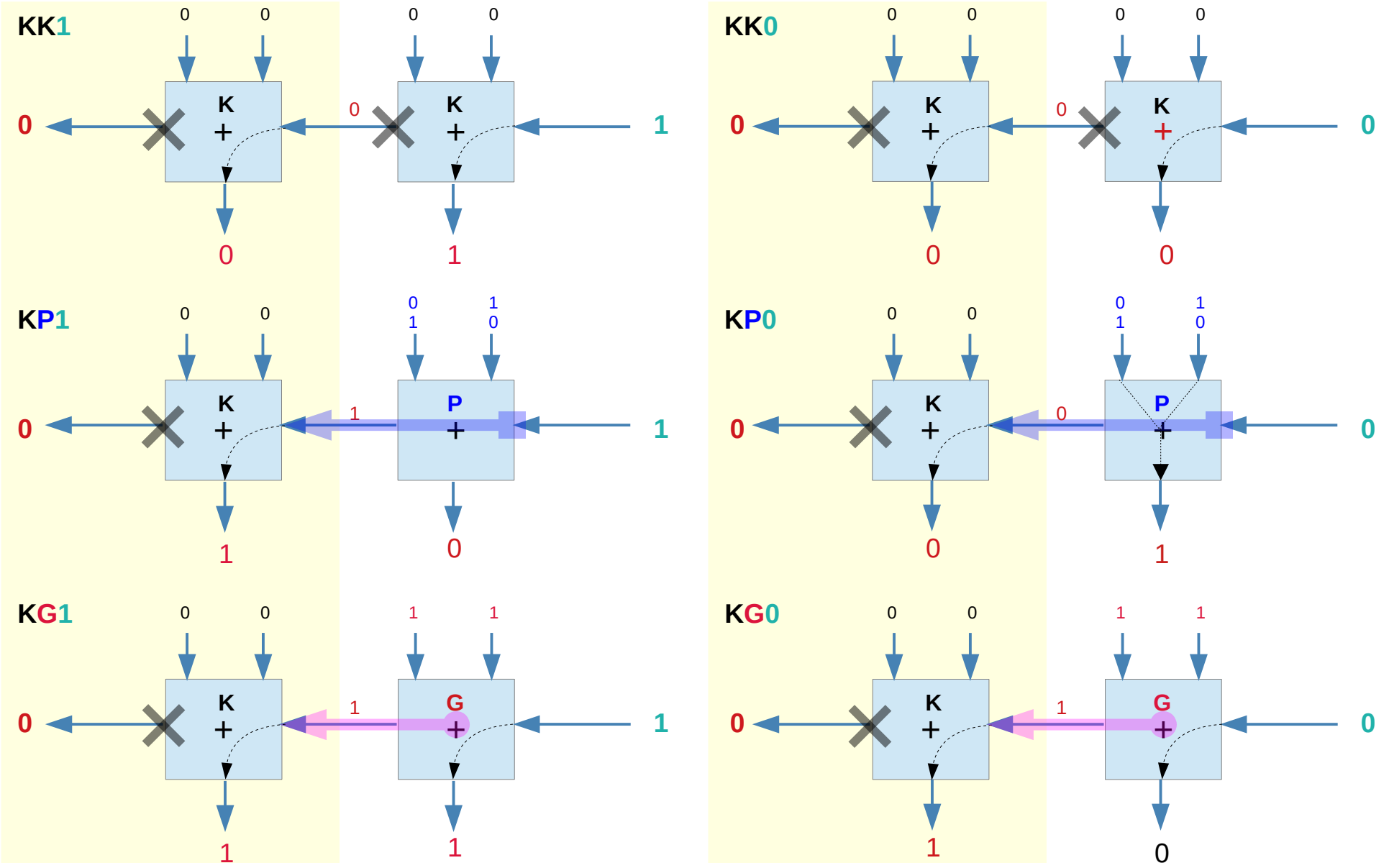
K, P, and G conditions in a 2-bit adder (2)

X	Y		
0	0	K	Kill (=$\overline{P}\overline{G}$)
0	1	P	Propagate
1	0	P	Propagate
1	1	G	Generate

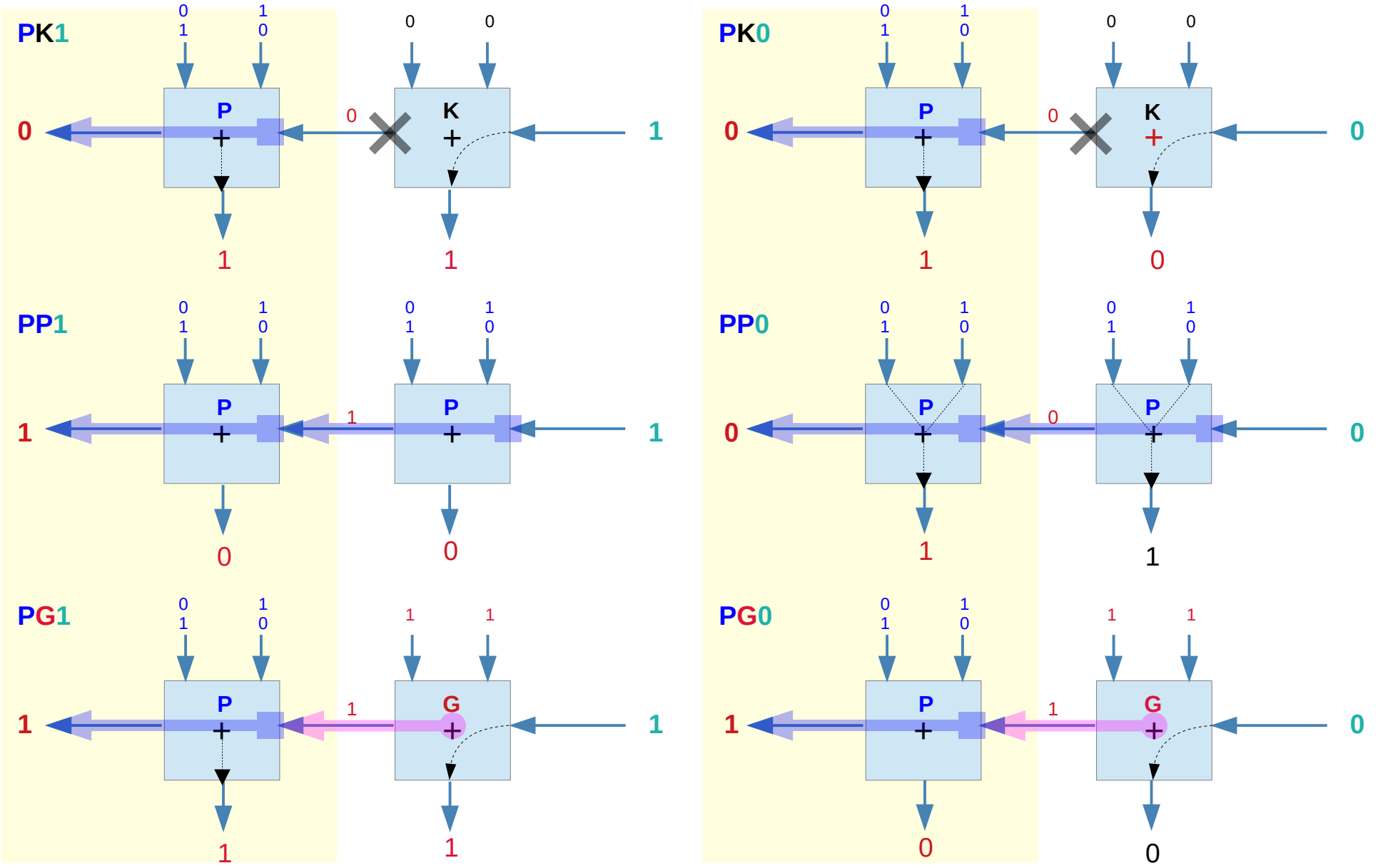


K	K	0
K	K	1
K	P	0
K	P	1
K	G	0
K	G	1
P	K	0
P	K	1
P	P	0
P	P	1
P	G	0
P	G	1
G	K	0
G	K	1
G	P	0
G	P	1
G	G	0
G	G	1

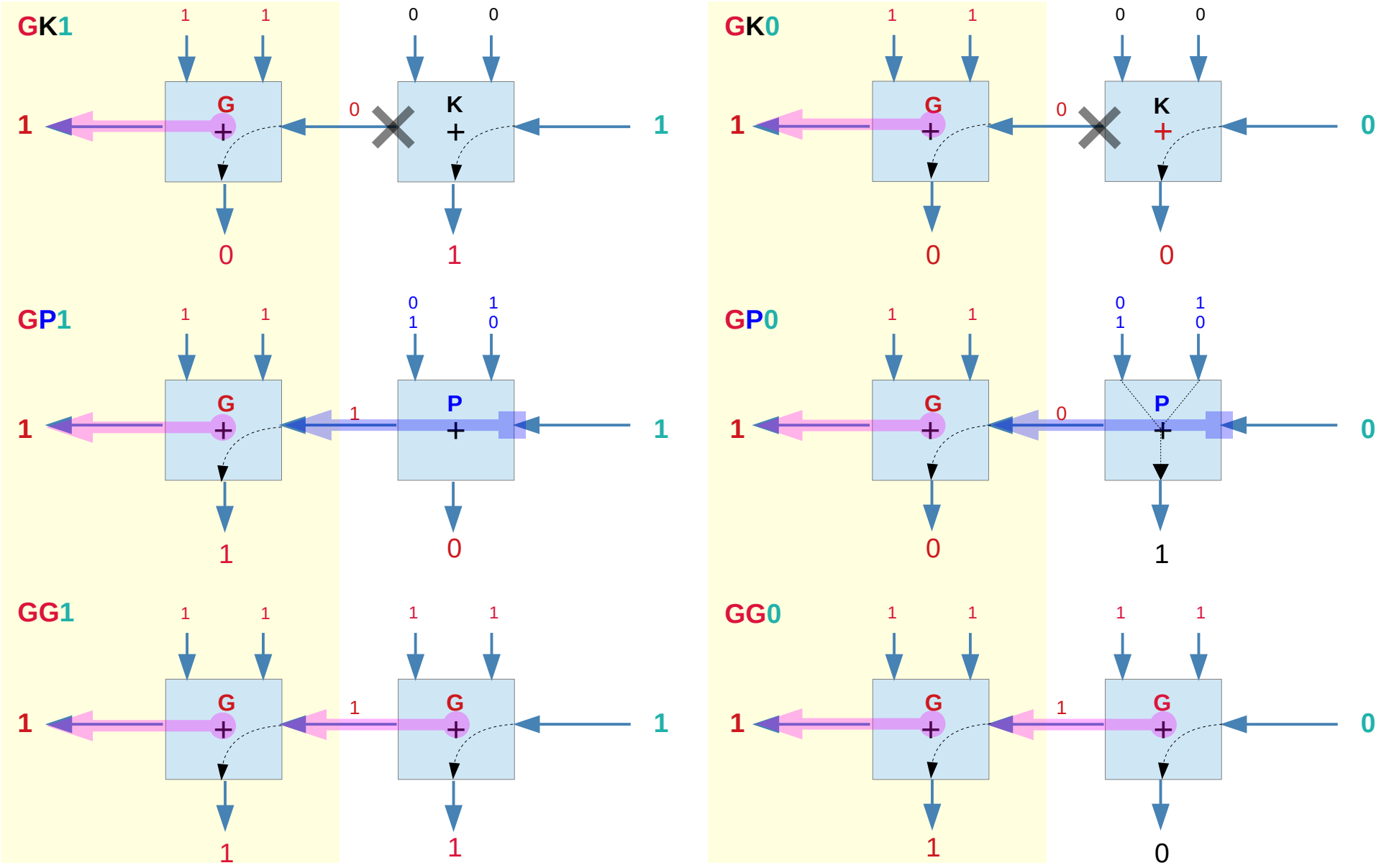
1. Cases when **FA1** is in the **K** mode



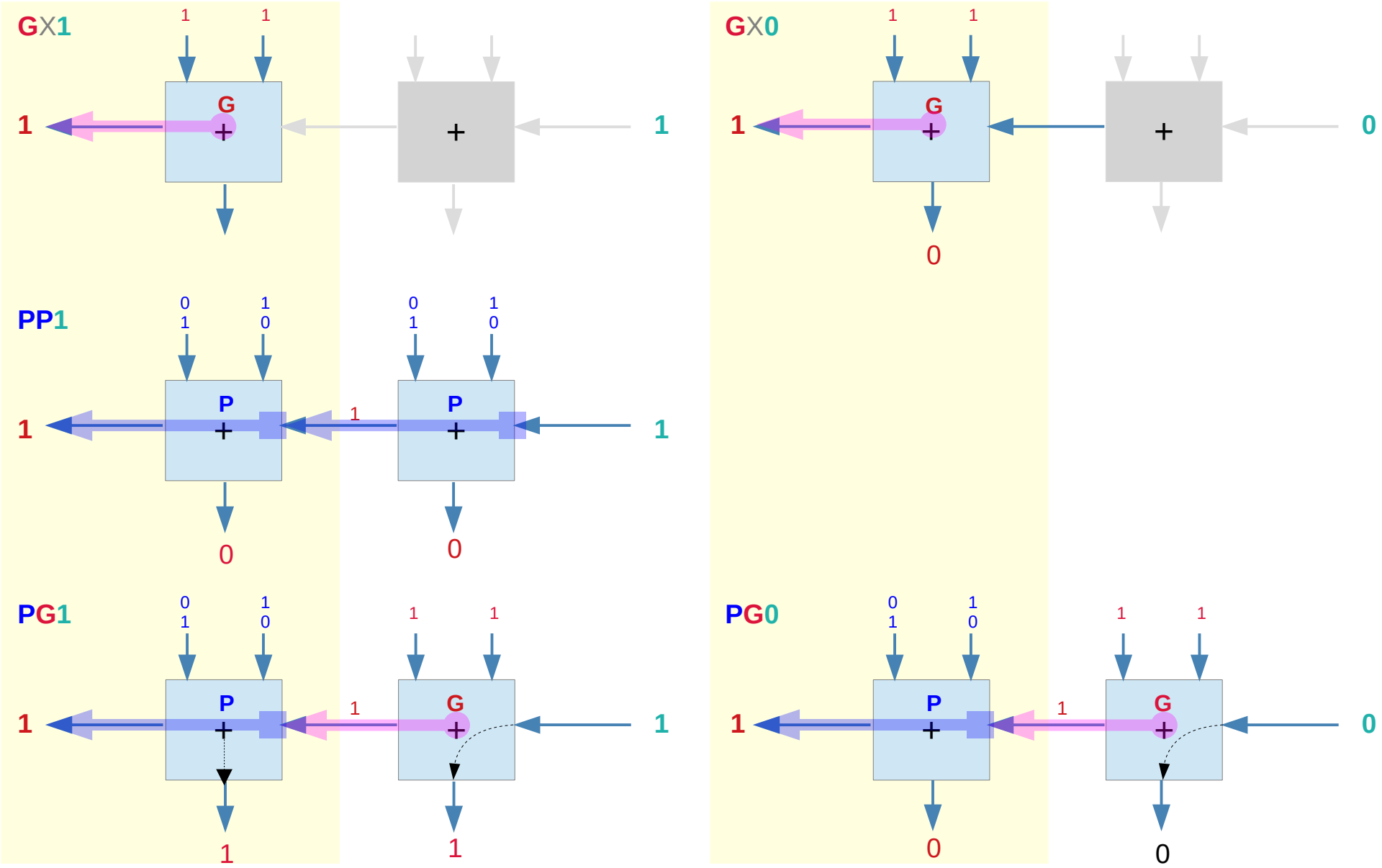
2. Cases when **FA1** is in the **P** mode



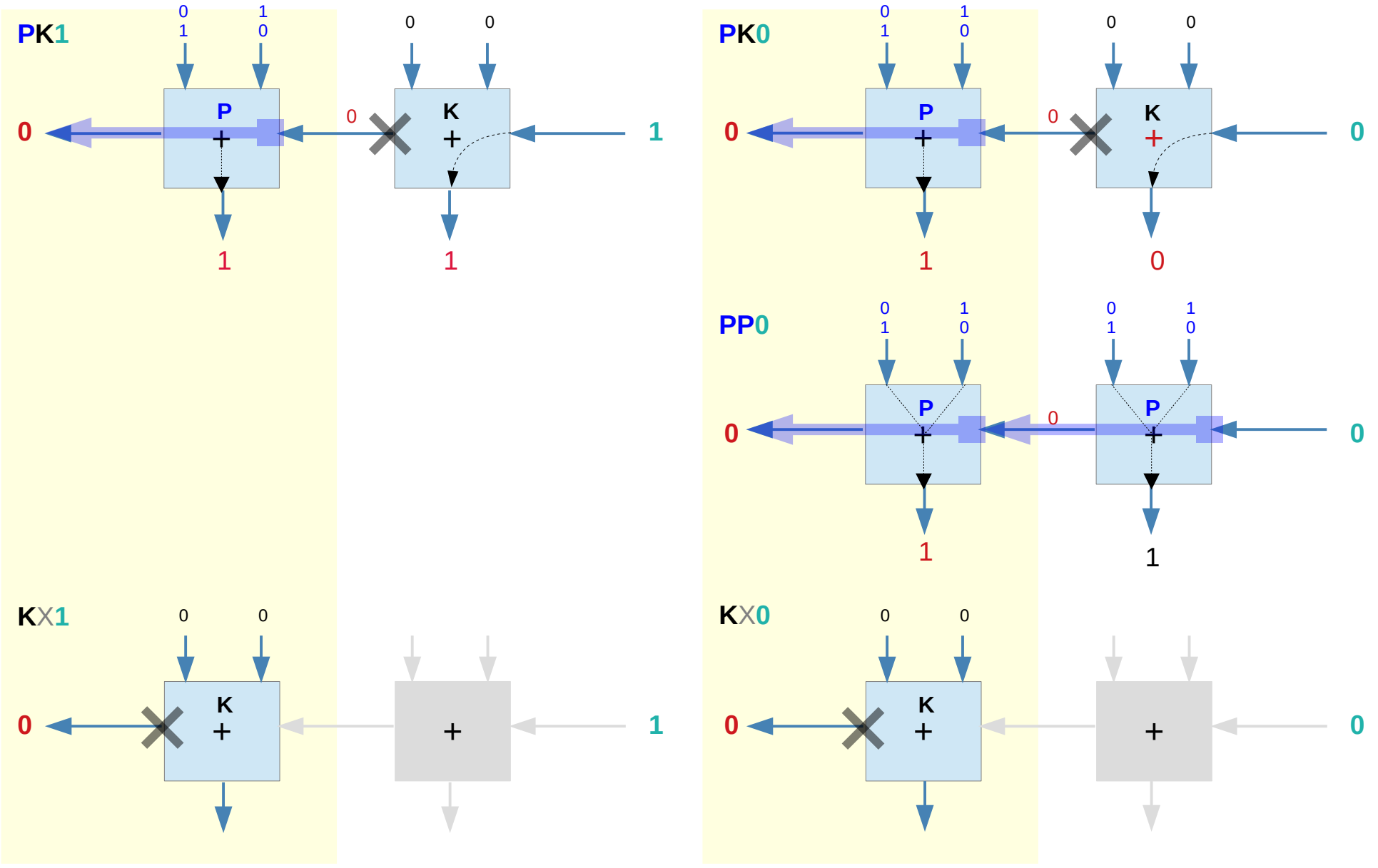
3. Cases when **FA1** is in the **G** mode



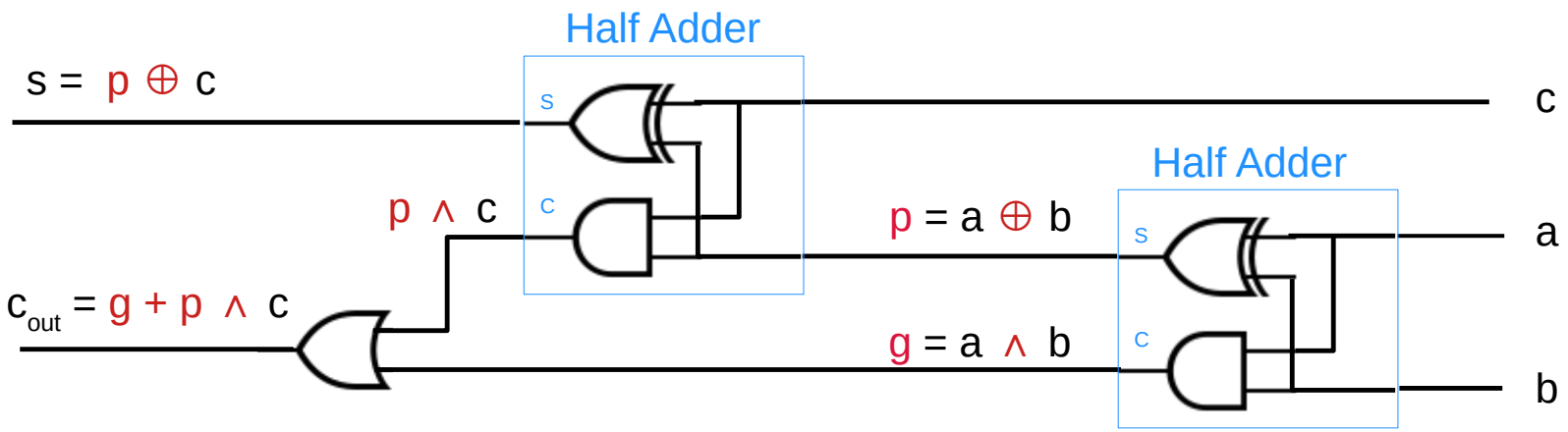
Cases for $C_{out} = 1$



Cases for $C_{out} = 0$

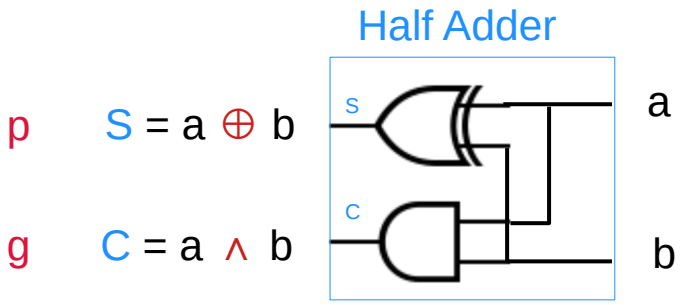


FA with P & G



Half Adder
S = a ⊕ b
C = a ∧ b

a	b	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0



Full adder with additional generate and propagate signals.

https://en.wikipedia.org/wiki/Carry-skip_adder

Ripple Carry Adder

$p_i = a_i \oplus b_i$

$g_i = a_i \wedge b_i$

$c_1 = g_0 + p_0 \wedge c_0$

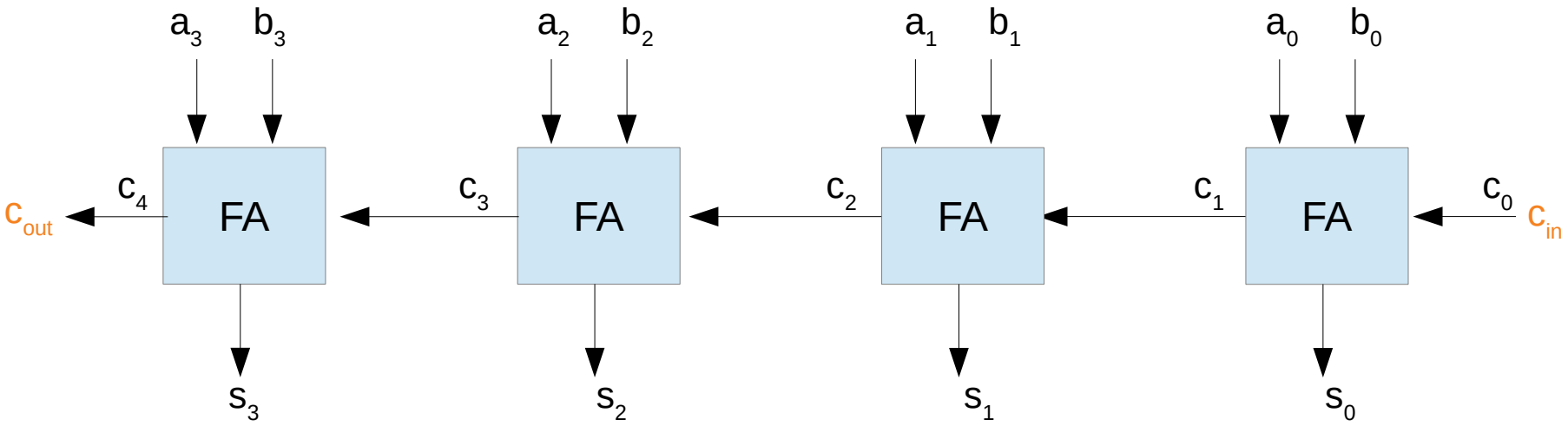
$c_2 = g_1 + p_1 \wedge c_1$

$c_3 = g_2 + p_2 \wedge c_2$

$c_4 = g_3 + p_3 \wedge c_3$

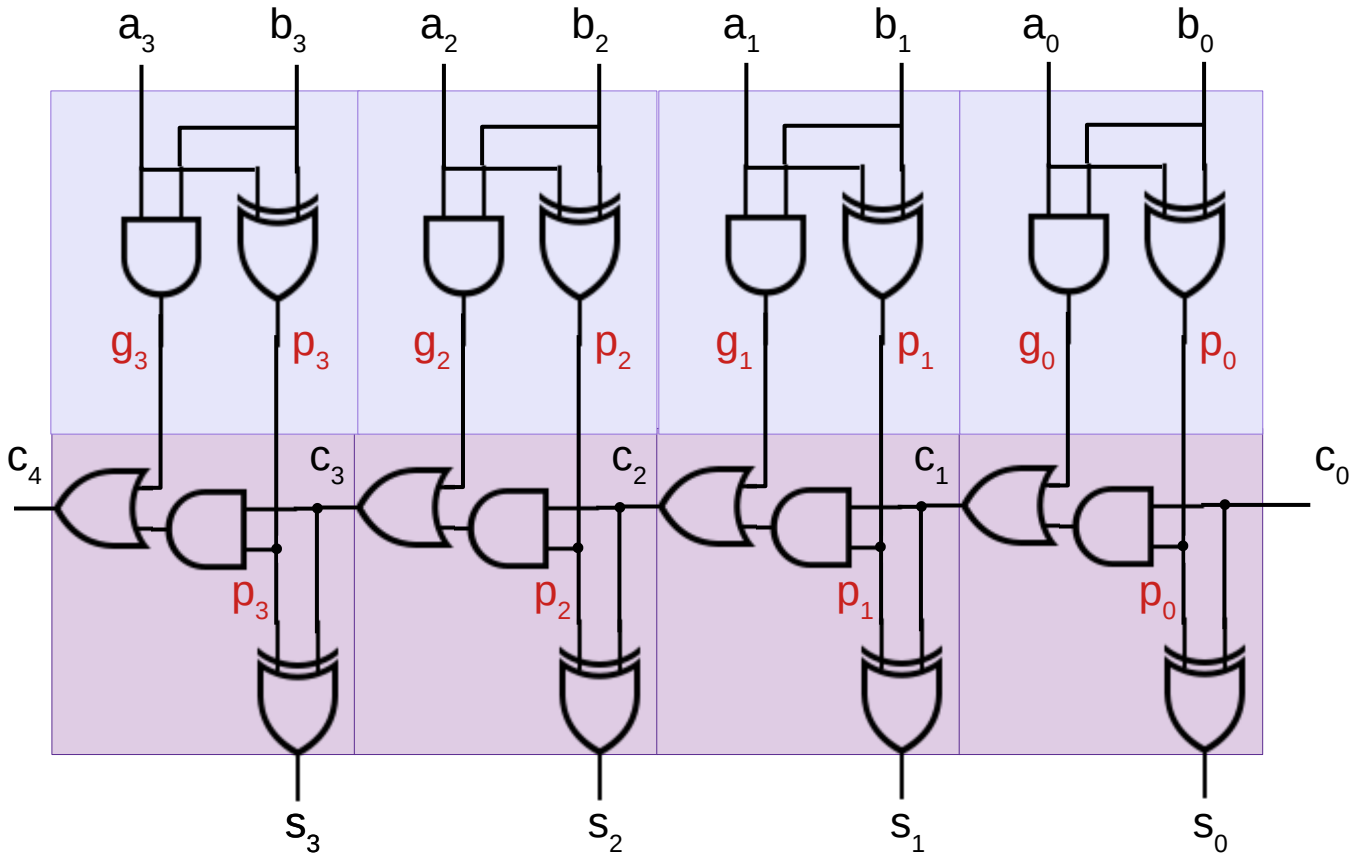
generated carry

propagated carry



https://en.wikipedia.org/wiki/Carry-skip_adder

4-bit Full Adder with P and G



Half Adder

$p_i = a_i \oplus b_i$

$g_i = a_i \wedge b_i$

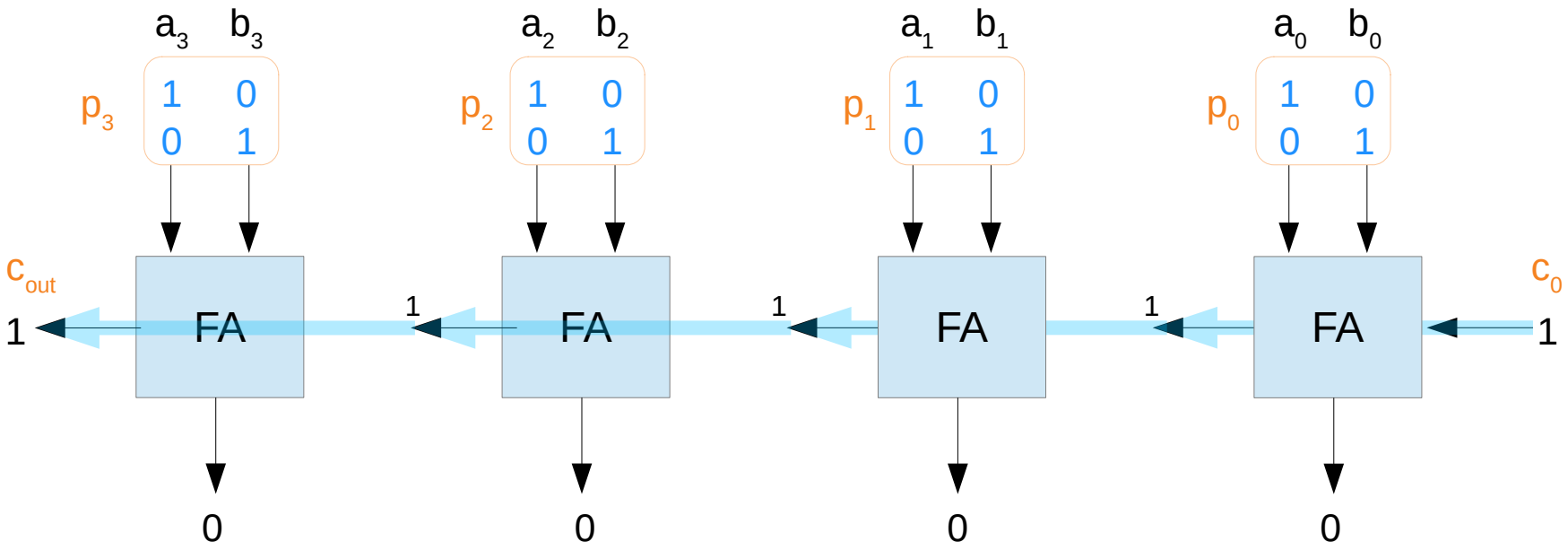
Half Adder

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

$s_i = p_i \oplus c_i$

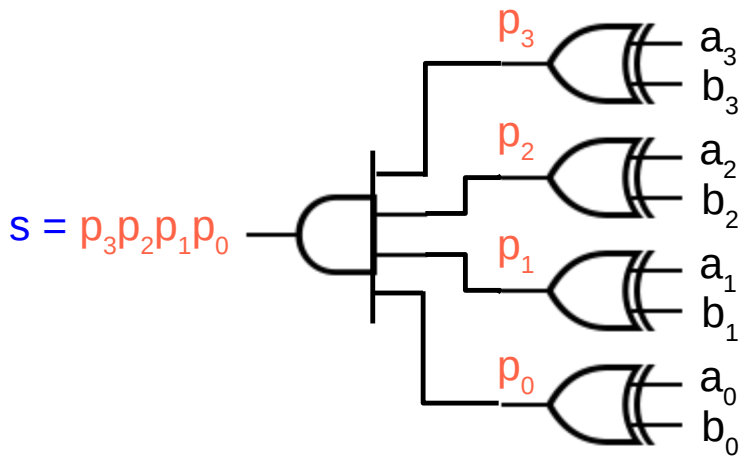
https://upload.wikimedia.org/wikiversity/en/1/18/RCA.Note.H.1.20151215.pdf

C_0 propagation condition



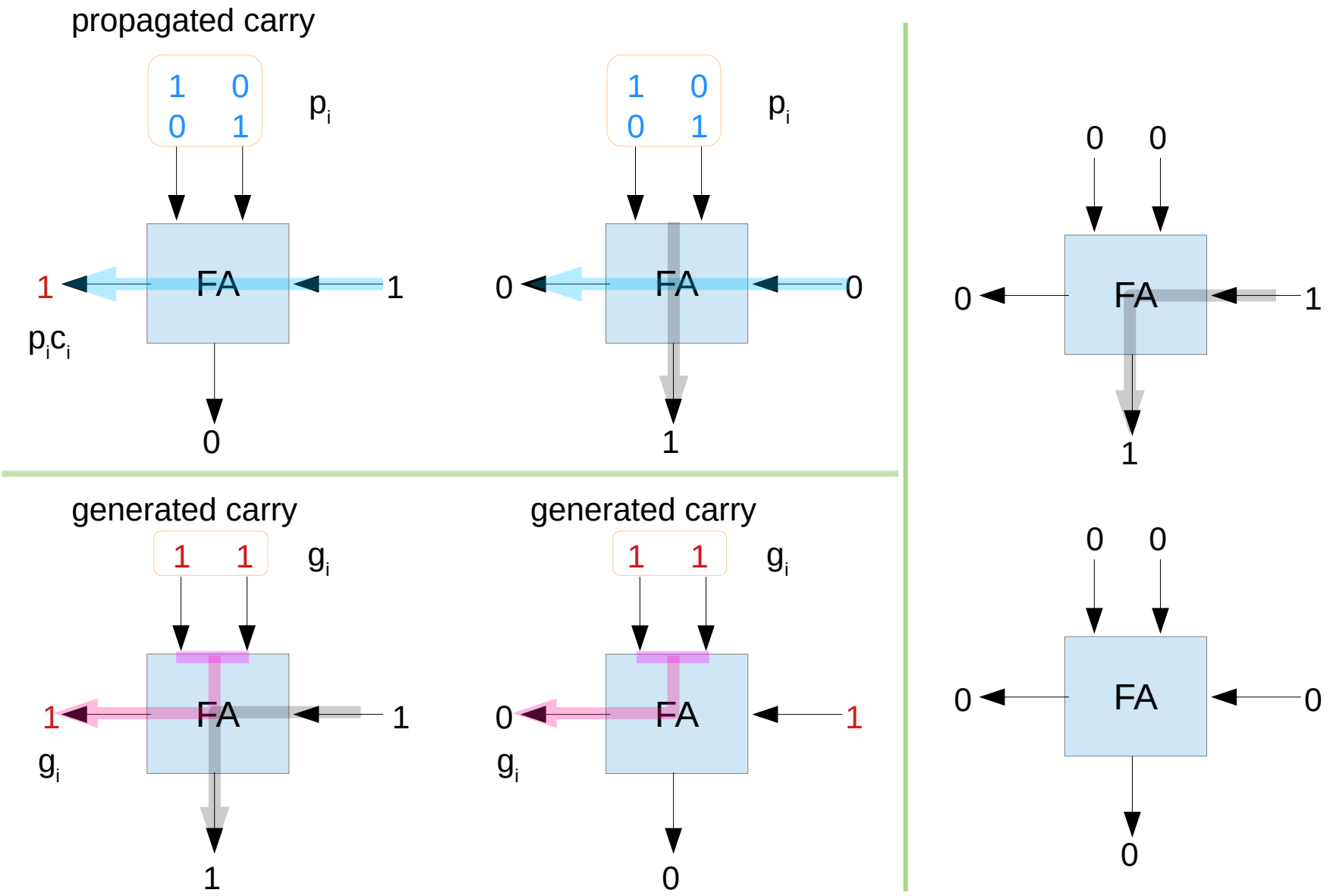
c_0 can be propagated to c_{out} only when $s = 1$

$$s = p_3 \wedge p_2 \wedge p_1 \wedge p_0 = p_{[3:0]}$$
$$= (a_3 \oplus b_3) \wedge (a_2 \oplus b_2) \wedge (a_1 \oplus b_1) \wedge (a_0 \oplus b_0)$$



https://en.wikipedia.org/wiki/Carry-skip_adder

Propagated and Generated Carries



Invert

Invert

$(C1_3 \oplus C1_2)$

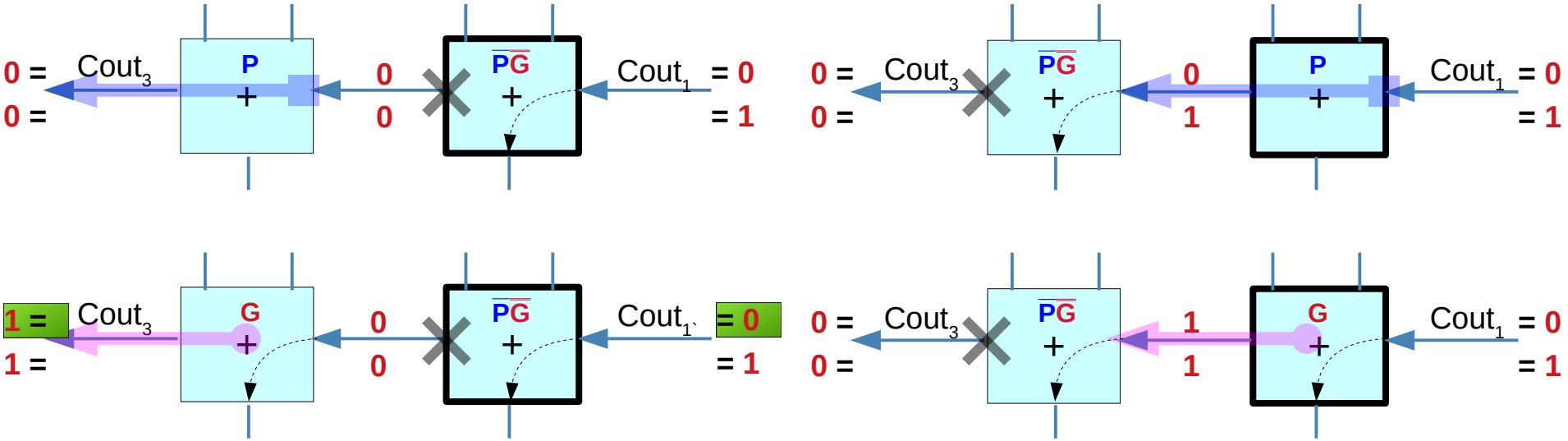
P	P
G	G
P	P'G'
G	P'G'
P'G'	P
P'G'	G

$C1 = X + Y$

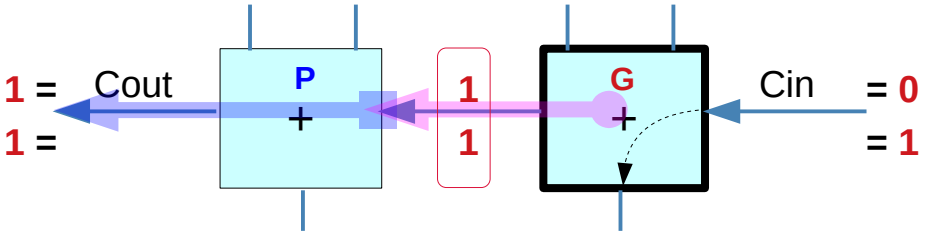
$C0 = X \cdot Y$

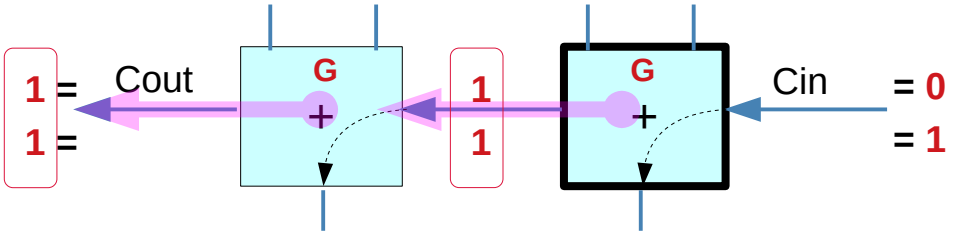
$C1 \oplus C0 = X \oplus Y$ = P

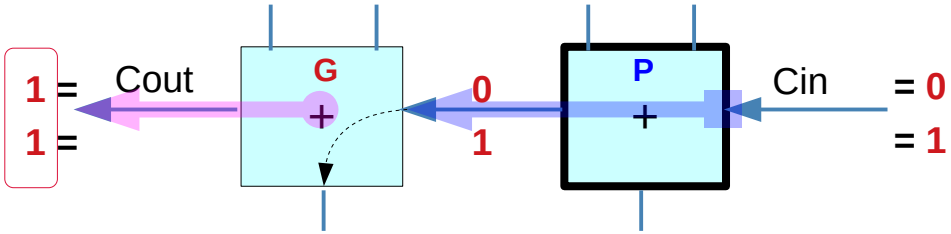
P	G
G	

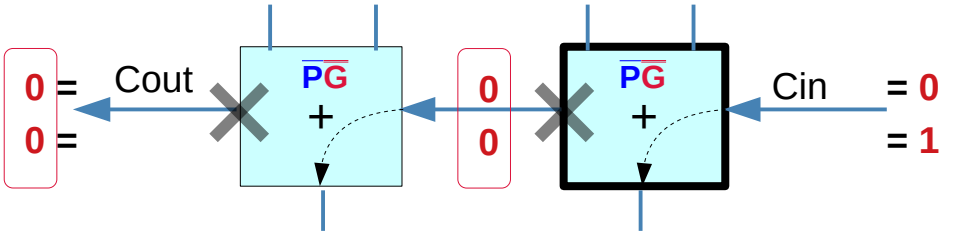


Variable Block

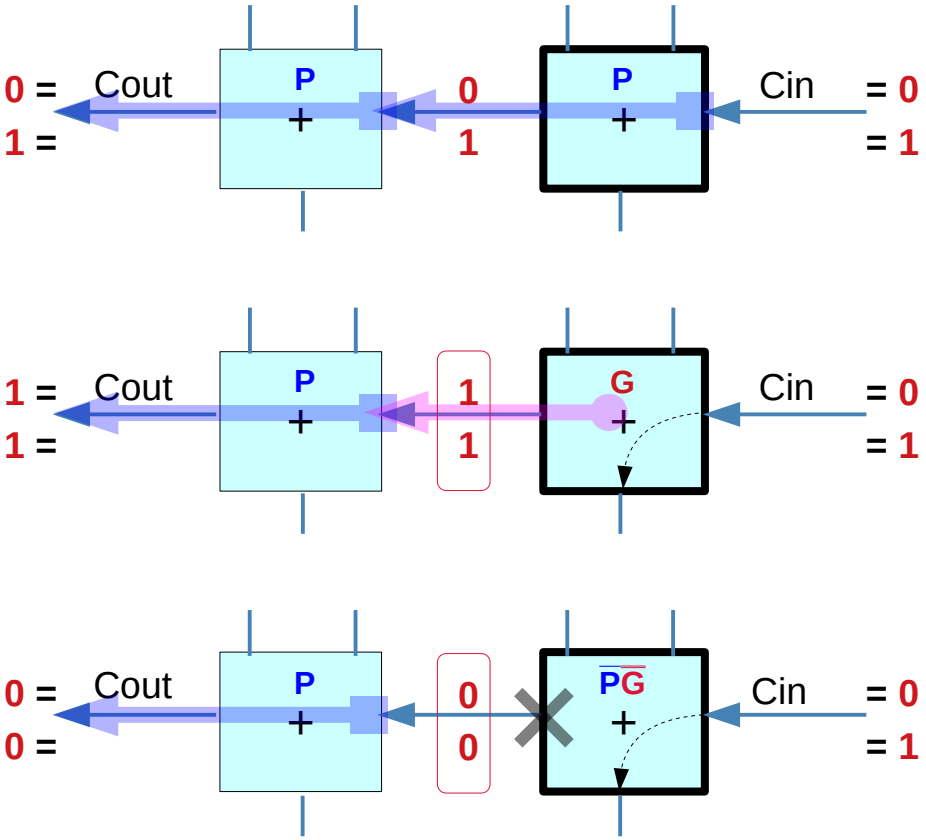








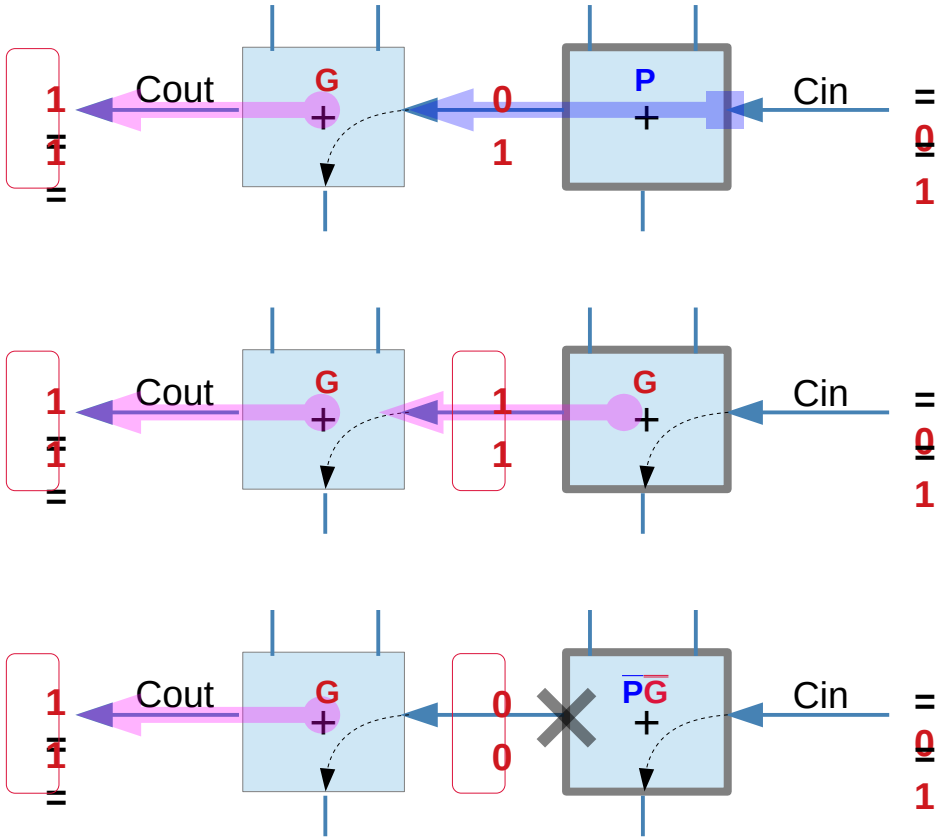
Variable Block



Propagate

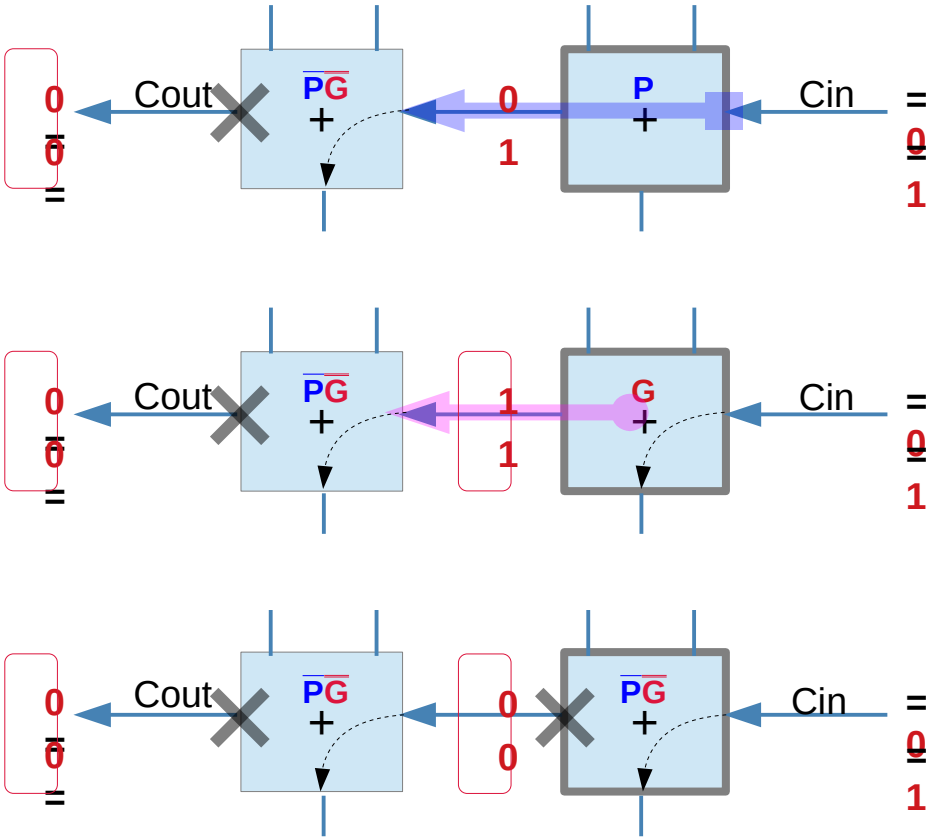
Invert

Variable Block

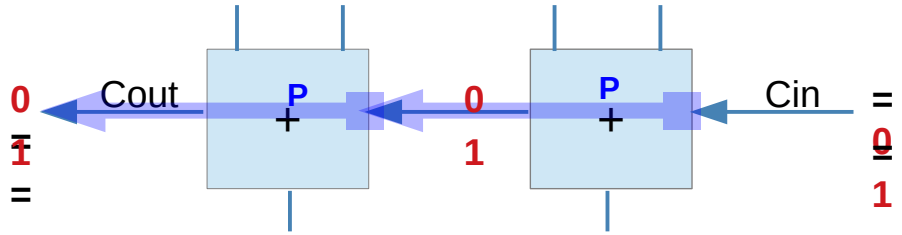


Invert

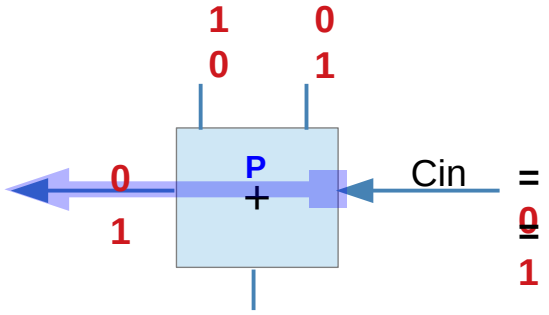
Variable Block

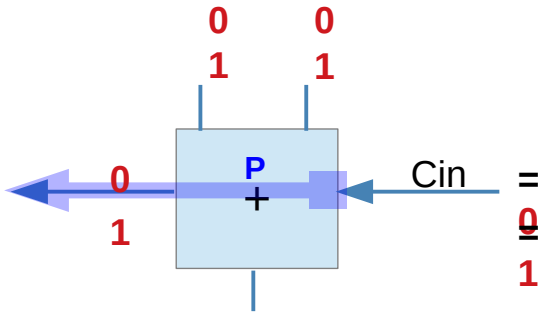


Variable Block

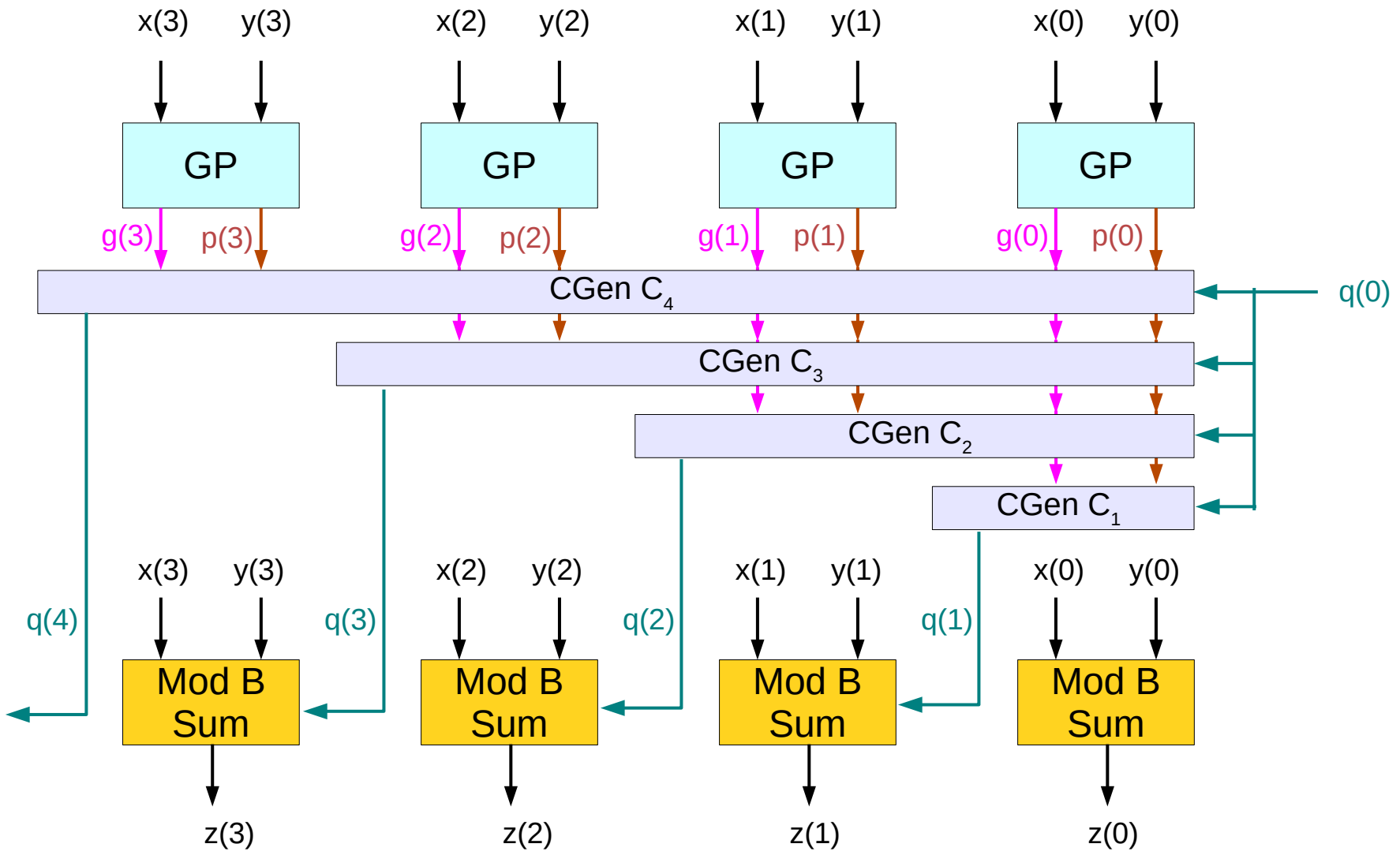


Propagate



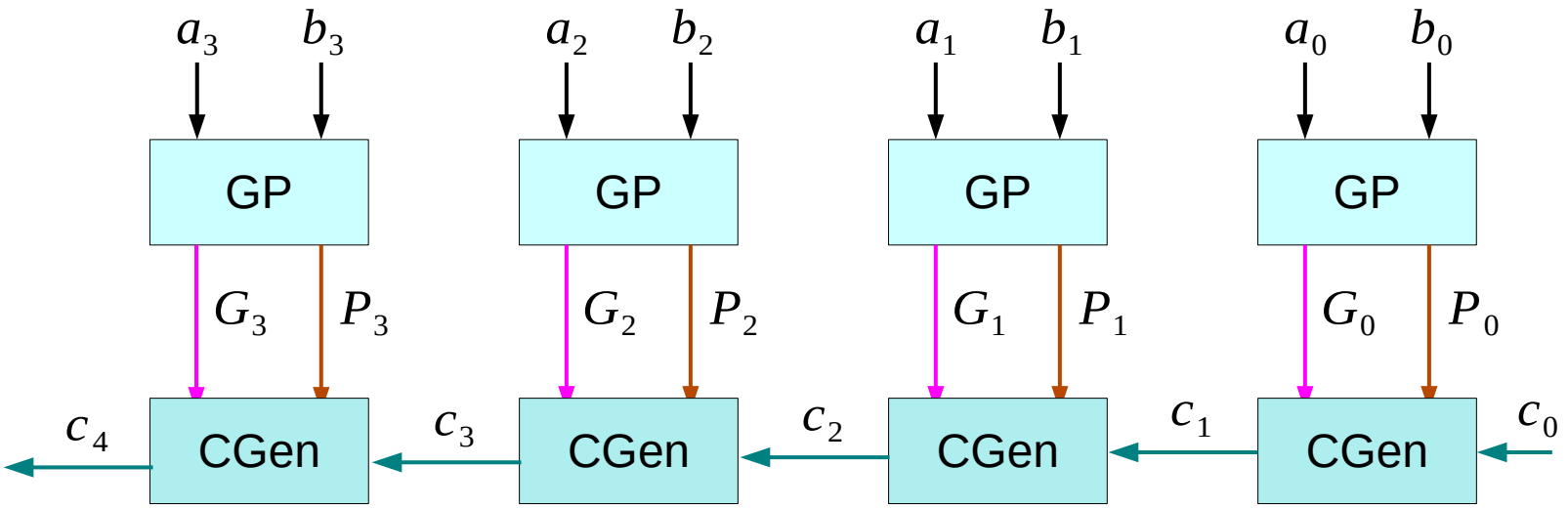


Carry Lookahead Adder



Synthesis of Arithmetic Circuits: FPGA, ASIC and Ebedded Systems, J-P Deschamps et al

Carry Lookahead Adder

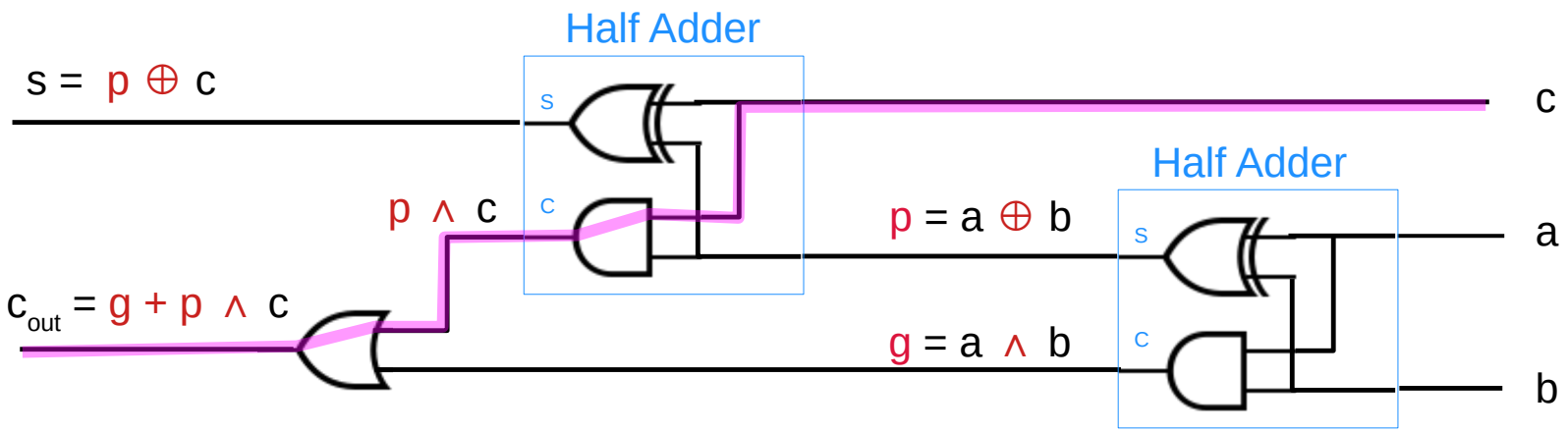


$$c_4 = G_3 + P_3 c_3 \qquad c_3 = G_2 + P_2 c_2 \qquad c_2 = G_1 + P_1 c_1 \qquad c_1 = G_0 + P_0 c_0$$

$$\begin{aligned} G_0 + P_0 c_0 &= c_1 \\ G_1 + P_1 G_0 + P_1 P_0 c_0 &= c_2 \\ G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 c_0 &= c_3 \\ G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 c_0 &= c_4 \end{aligned}$$

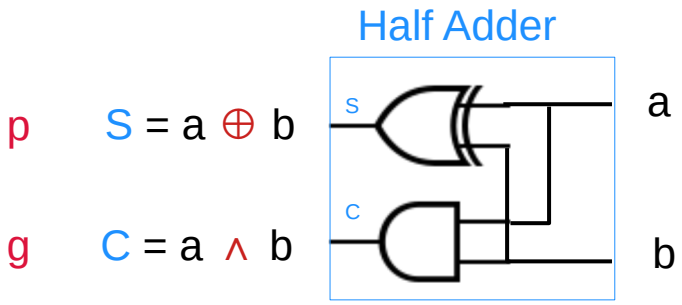
Synthesis of Arithmetic Circuits: FPGA, ASIC and Ebedded Systems, J-P Deschamps et al

FA with P & G



Half Adder
S = a ⊕ b
C = a ∧ b

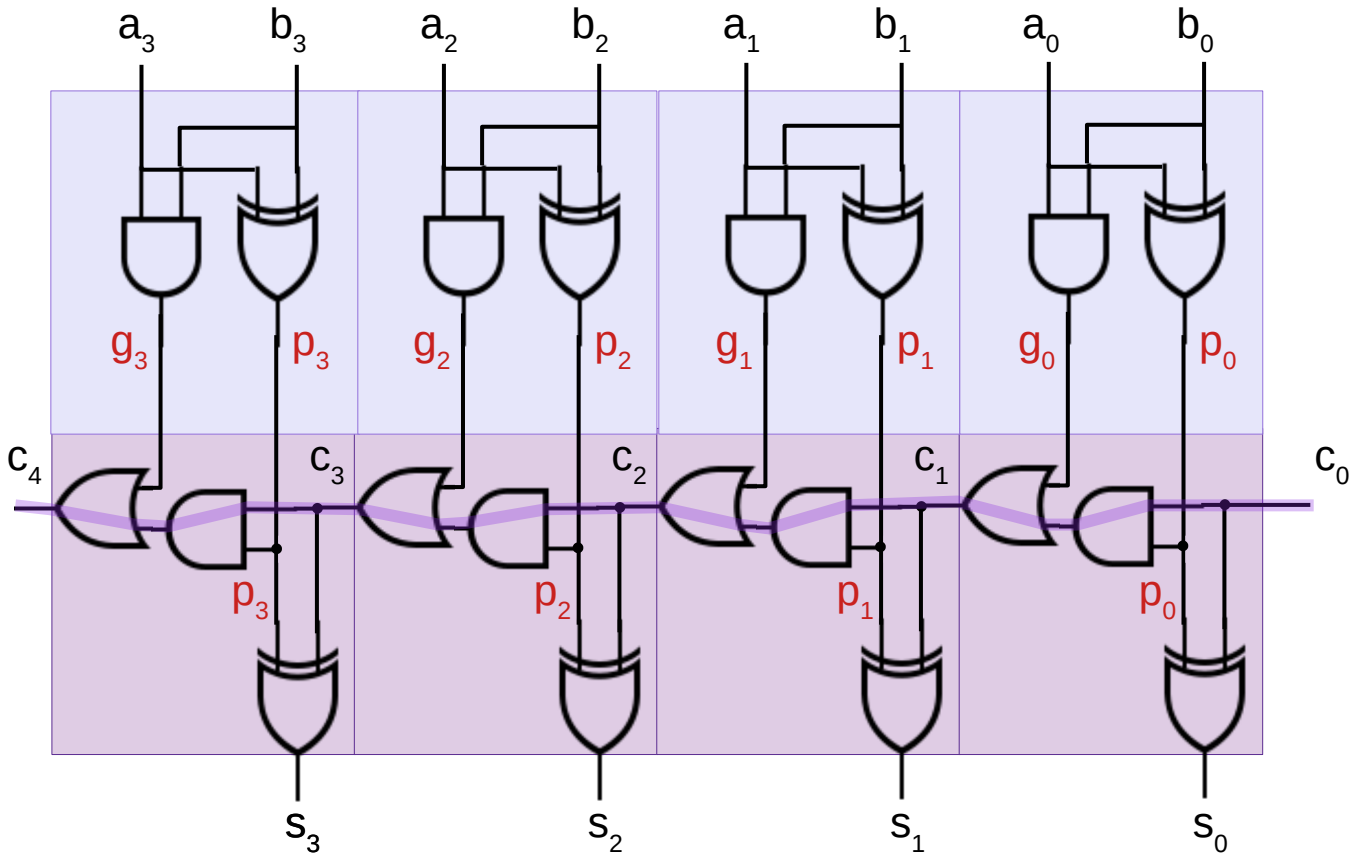
a	b	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0



https://en.wikipedia.org/wiki/Carry-skip_adder

Full adder with additional generate and propagate signals.

4-bit Full Adder with P and G



Half Adder

$p_i = a_i \oplus b_i$

$g_i = a_i \wedge b_i$

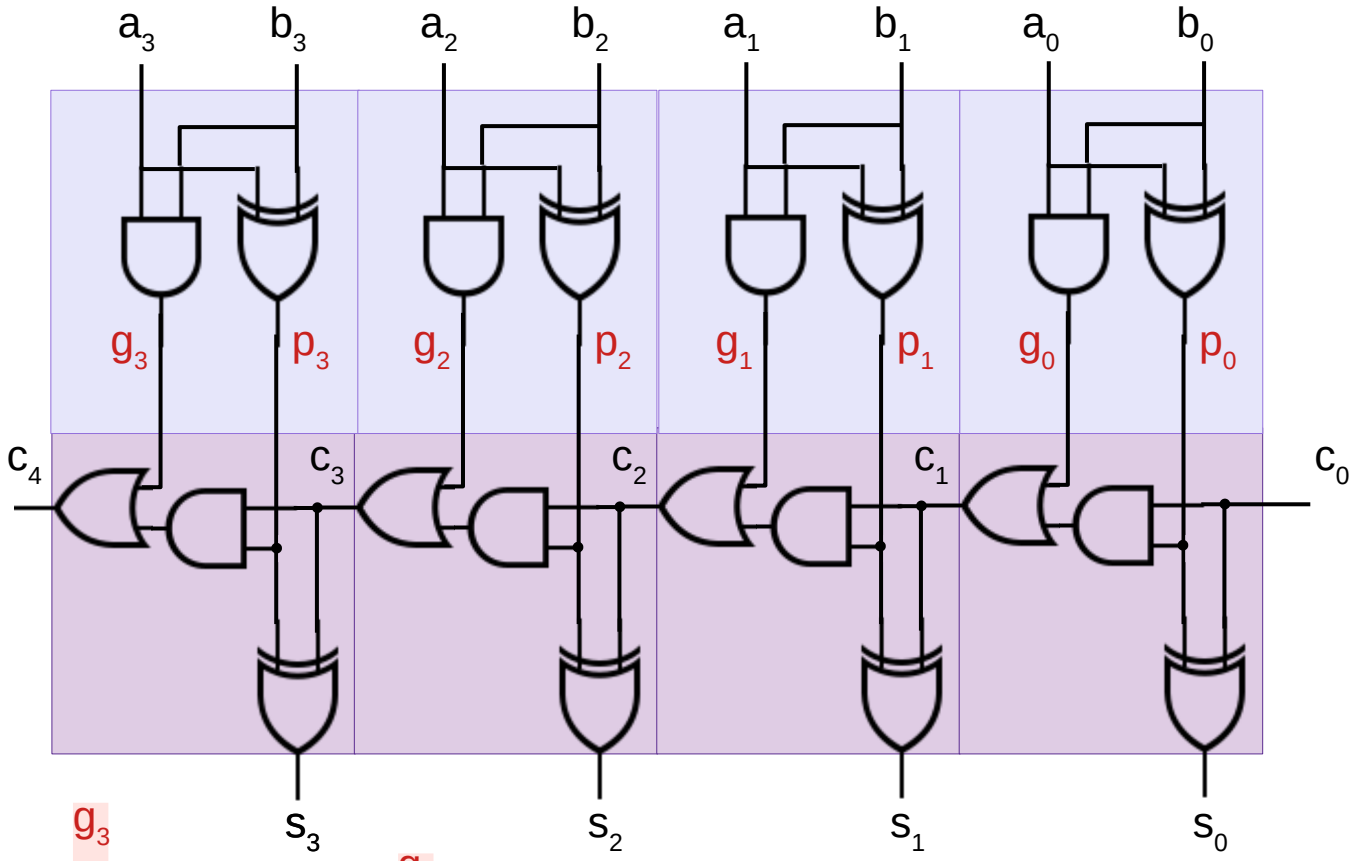
Half Adder

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

$s_i = p_i \oplus c_i$

https://upload.wikimedia.org/wikiversity/en/1/18/RCA.Note.H.1.20151215.pdf

4-bit Full Adder with P and G



Half Adder

$p_i = a_i \oplus b_i$

$g_i = a_i \wedge b_i$

Half Adder

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

$s_i = p_i \oplus c_i$

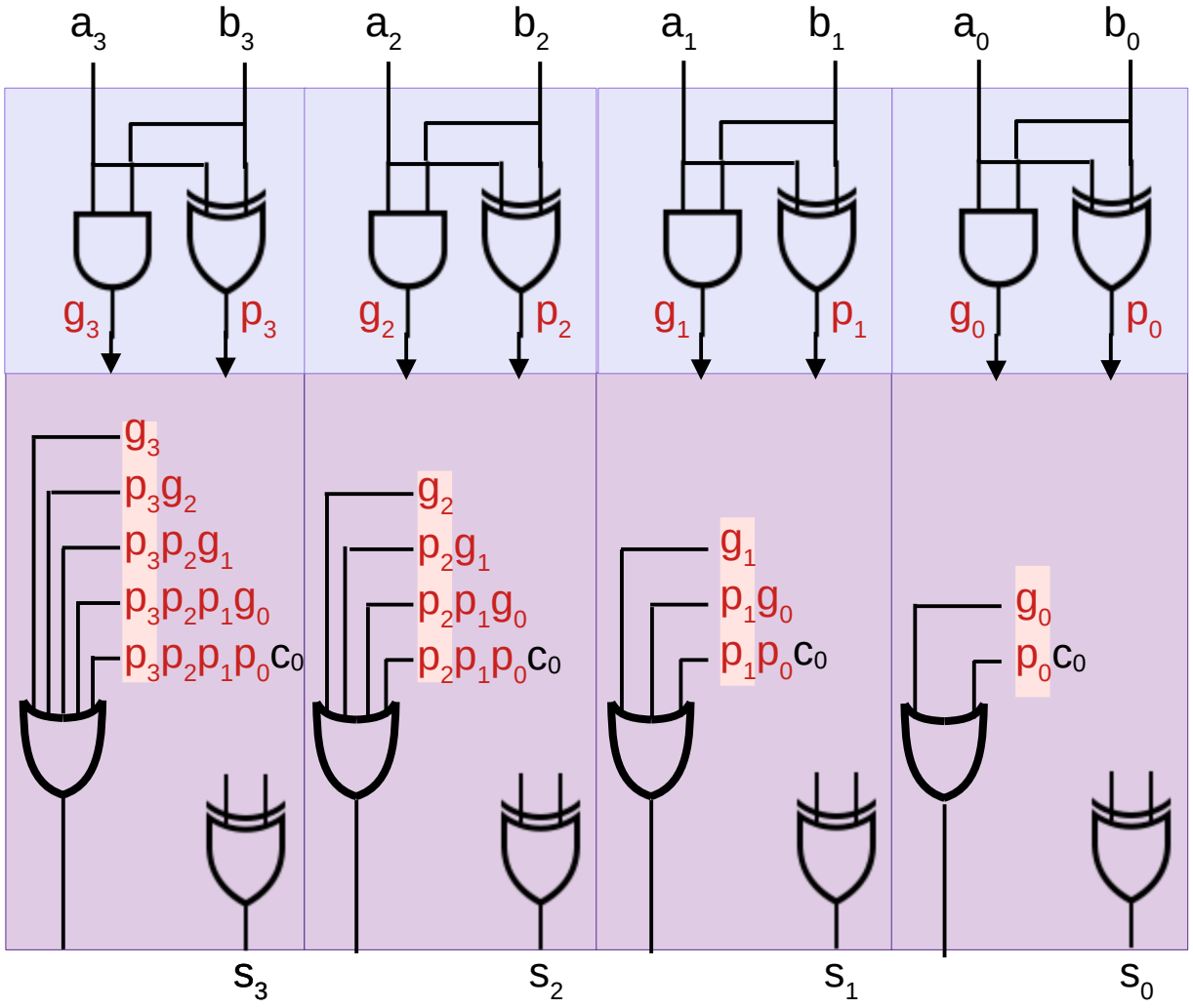
g_3
$p_3 g_2$
$p_3 p_2 g_1$
$p_3 p_2 p_1 g_0$
$p_3 p_2 p_1 p_0 c_0$

g_2
$p_2 g_1$
$p_2 p_1 g_0$
$p_2 p_1 p_0 c_0$

g_1
$p_1 g_0$
$p_1 p_0 c_0$

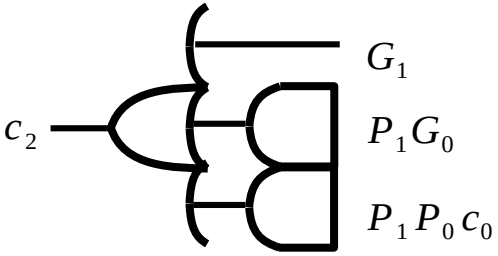
g_0
$p_0 c_0$

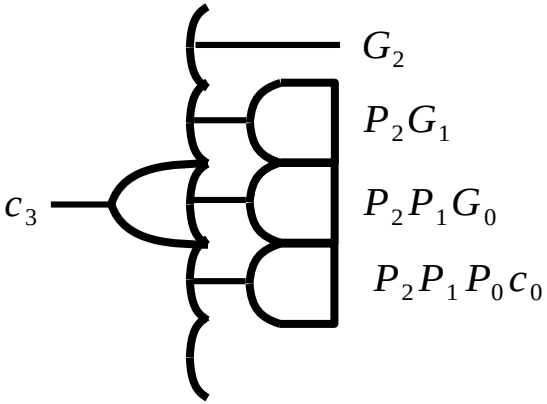
4-bit Full Adder with P and G

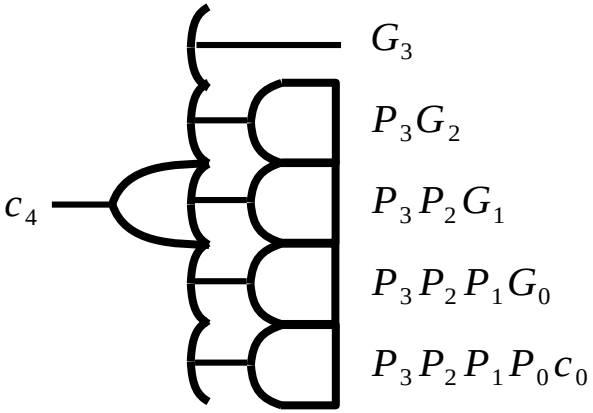


Carry Equations into Gates (2)

$$G_0 + P_0 c_0 = c_1$$
$$G_1 + P_1 G_0 + P_1 P_0 c_0 = c_2$$
$$G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 c_0 = c_3$$
$$G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 c_0 = c_4$$







Fan-in number: large
Stage number: small

Carry Equations

$$c_{i+1} = G_i + P_i c_i$$

$$c_1 = G_0 + P_0 c_0 \qquad c_1 = G_0 + P_0 c_0$$

$$c_2 = G_1 + P_1 c_1 \qquad c_2 = G_1 + P_1 [G_0 + P_0 c_0]$$

$$c_3 = G_2 + P_2 c_2 \qquad c_3 = G_2 + P_2 [G_1 + P_1 [G_0 + P_0 c_0]]$$

$$c_4 = G_3 + P_3 c_3 \qquad c_4 = G_3 + P_3 [G_2 + P_2 [G_1 + P_1 [G_0 + P_0 c_0]]]$$

$$G_0 + P_0 c_0 = c_1$$

$$G_1 + P_1 G_0 + P_1 P_0 c_0 = c_2$$

$$G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 c_0 = c_3$$

$$G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 c_0 = c_4$$

Carry Equations into Gates (1)

$$c_{i+1} = G_i + P_i c_i$$

$$c_1 = G_0 + P_0 c_0$$

$$c_2 = G_1 + P_1 c_1$$

$$c_3 = G_2 + P_2 c_2$$

$$c_4 = G_3 + P_3 c_3$$

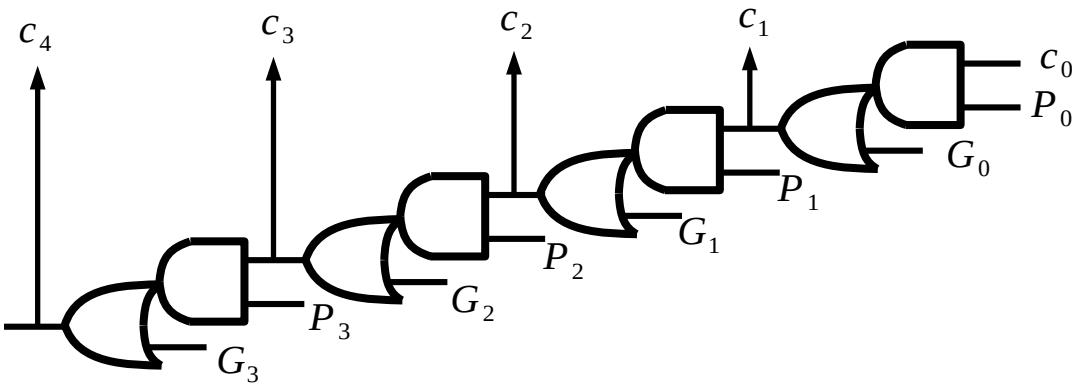
$$c_1 = G_0 + P_0 c_0$$

$$c_2 = G_1 + P_1 [G_0 + P_0 c_0]$$

$$c_3 = G_2 + P_2 [G_1 + P_1 [G_0 + P_0 c_0]]$$

$$c_4 = G_3 + P_3 [G_2 + P_2 [G_1 + P_1 [G_0 + P_0 c_0]]]$$

Fan-in number: small
Stage number: large



Full Carry Lookahead

AND2, OR2	$G_0 + P_0 c_0 = c_1$
AND3, OR3	$G_1 + P_1 G_0 + P_1 P_0 c_0 = c_2$
AND4, OR4	$G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 c_0 = c_3$
AND5, OR5	$G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 c_0 = c_4$
AND32, OR32	$= c_{31}$
AND33, OR33	$= c_{32}$

Large number of fan-in : Impractical

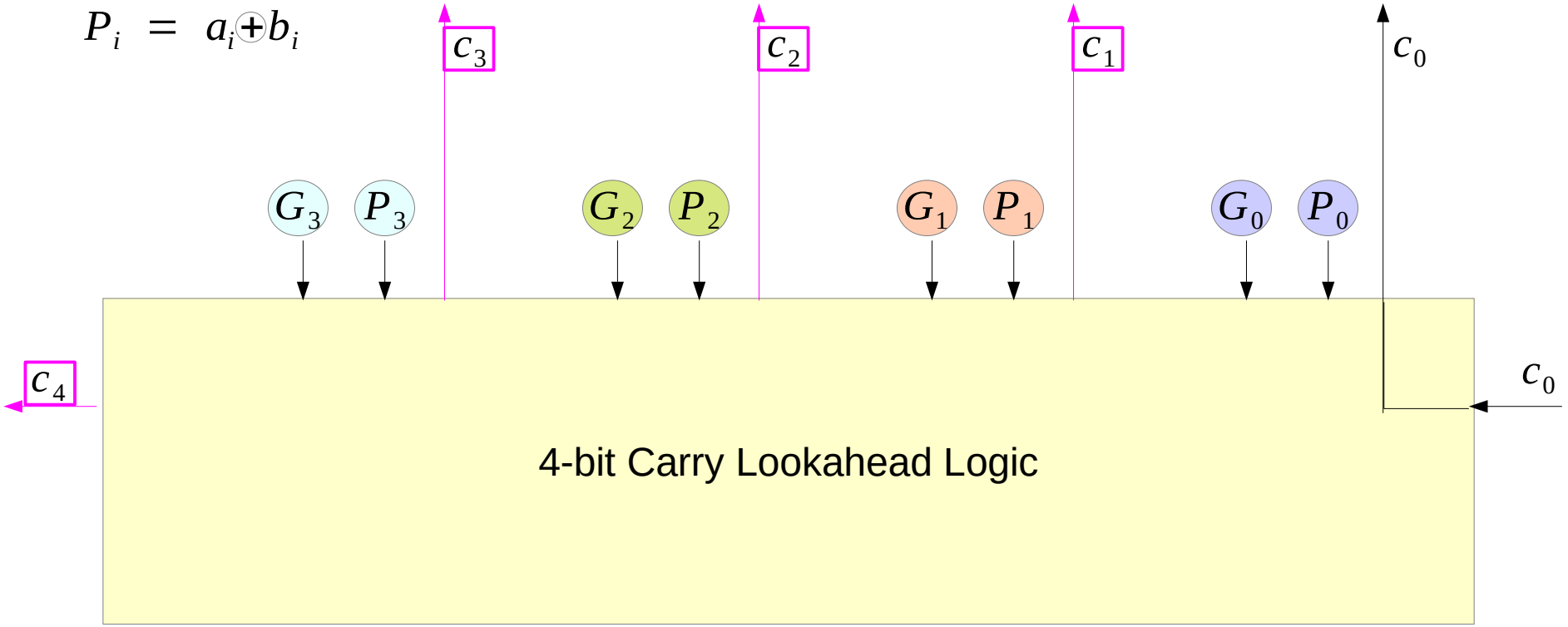
- High Radix Addition (2^g)
- Multi-level Lookahead

4-bit Carry Lookahead Logic – interface

All G_i's P_i's are computed simultaneously from a_i and b_i

each c_i's takes 2 gate delays

$$G_i = a_i \cdot b_i$$
$$P_i = a_i \oplus b_i$$

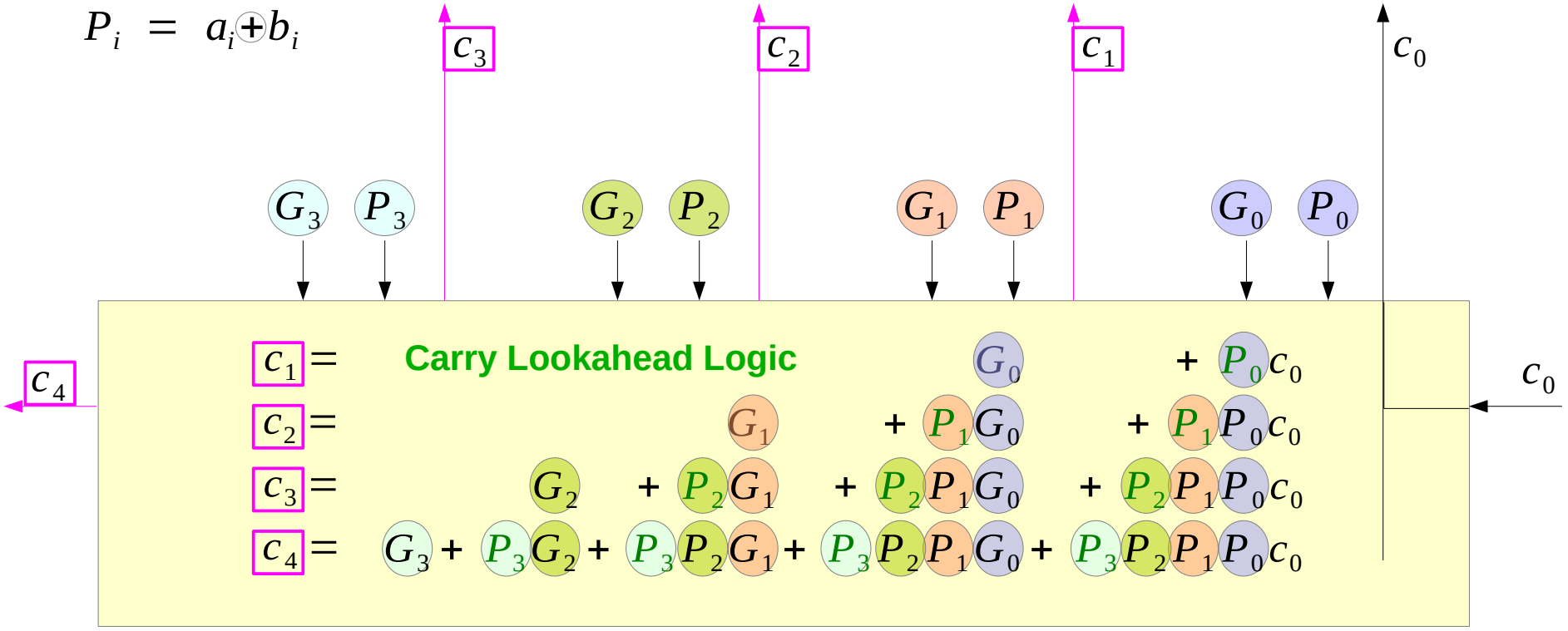


4-bit Carry Lookahead Logic – boolean equations

All G_i's P_i's are computed simultaneously from a_i and b_i

each c_i's takes 2 gate delays

$$G_i = a_i \cdot b_i$$
$$P_i = a_i \oplus b_i$$

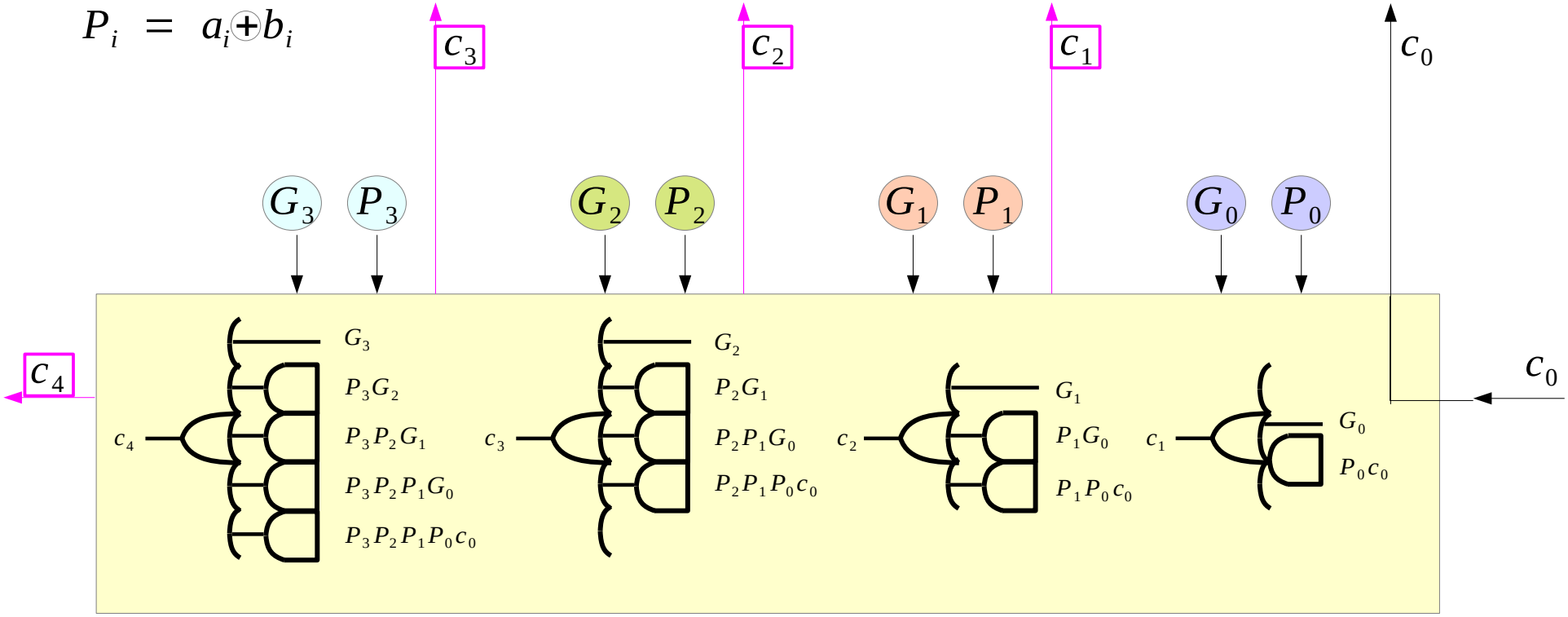


4-bit Carry Lookahead Logic – POS

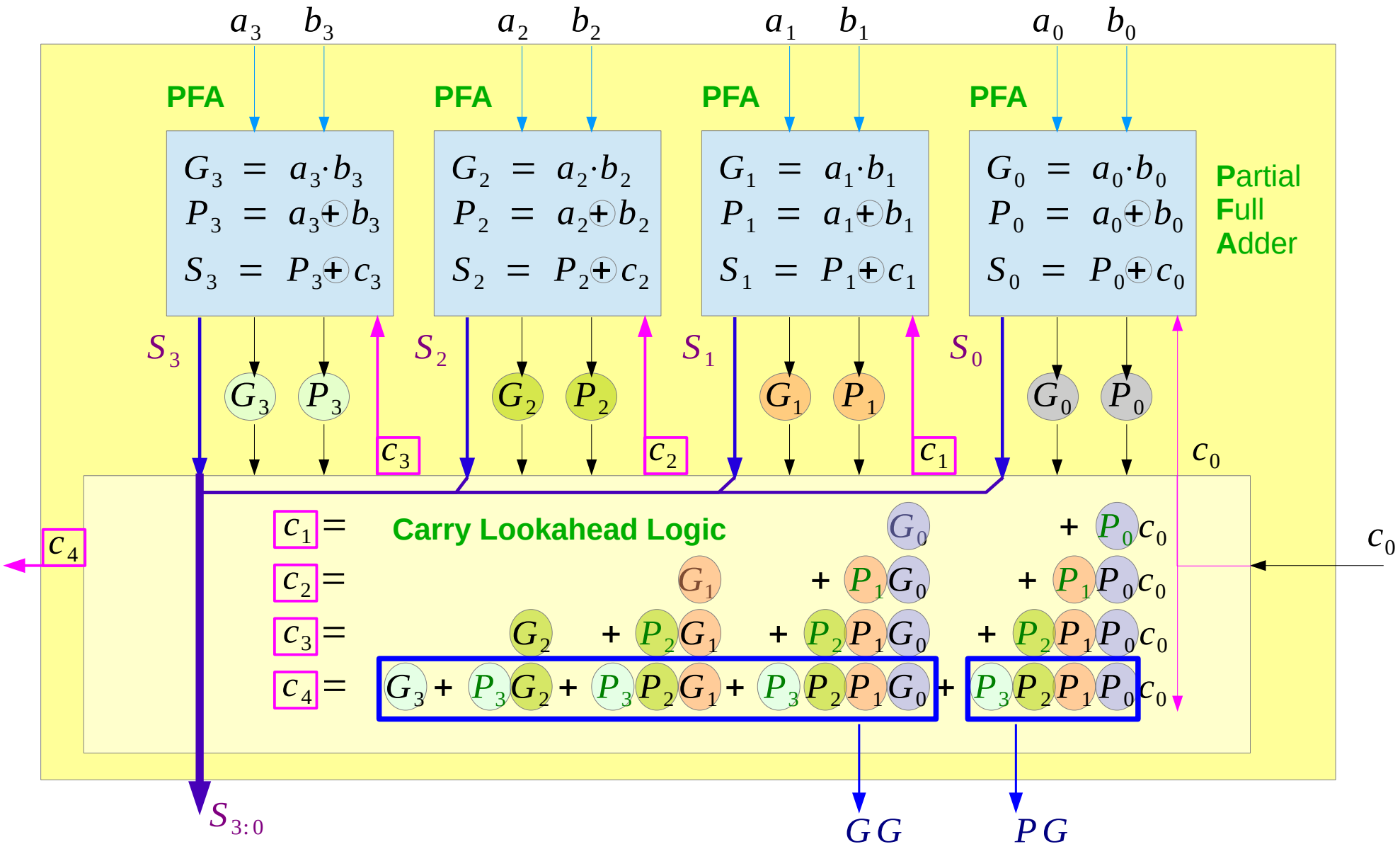
All G_i's P_i's are computed simultaneously from a_i and b_i

each c_i's takes 2 gate delays

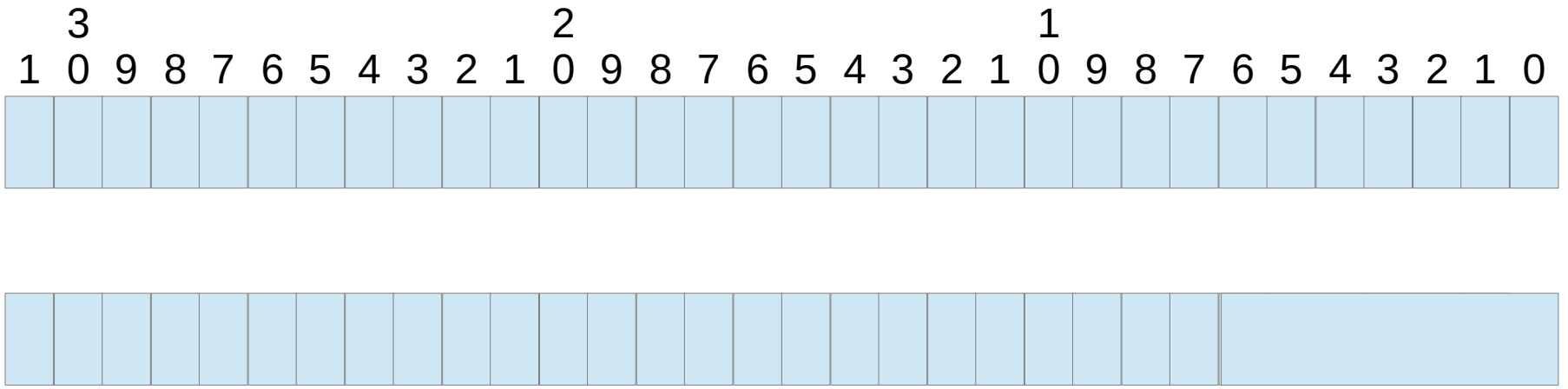
$$G_i = a_i \cdot b_i$$
$$P_i = a_i \oplus b_i$$



4-bit CLA


a_3 b_3 a_2 b_2 a_1 b_1 a_0 b_0
PFA
PFA
PFA
PFA
$G_3 = a_3 \cdot b_3$
$P_3 = a_3 \oplus b_3$
$S_3 = P_3 \oplus c_3$
$G_2 = a_2 \cdot b_2$
$P_2 = a_2 \oplus b_2$
$S_2 = P_2 \oplus c_2$
$G_1 = a_1 \cdot b_1$
$P_1 = a_1 \oplus b_1$
$S_1 = P_1 \oplus c_1$
$G_0 = a_0 \cdot b_0$
$P_0 = a_0 \oplus b_0$
$S_0 = P_0 \oplus c_0$
Partial Full Adder
S_3 S_2 S_1 S_0
G_3 P_3 G_2 P_2 G_1 P_1 G_0 P_0
c_3 c_2 c_1 c_0
c_4
Carry Lookahead Logic
$c_1 = G_0 + P_0 c_0$
$c_2 = G_1 + P_1 G_0 + P_1 P_0 c_0$
$c_3 = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 c_0$
$c_4 = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 c_0$
c_0
$S_{3:0}$
GG
PG

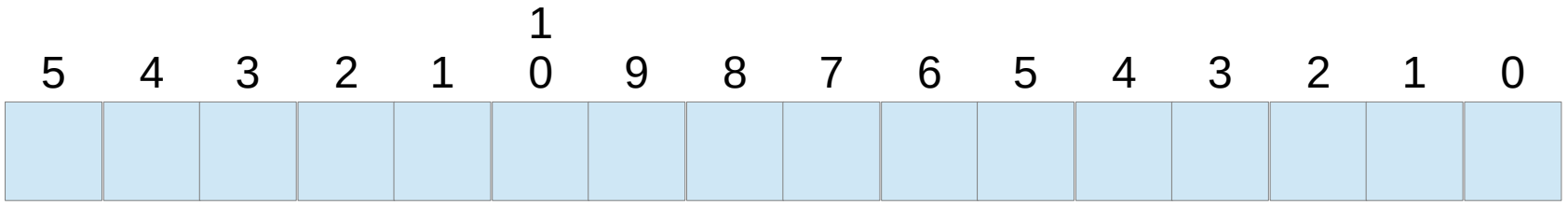
Multi-level Carry Lookahead



High Radix Addition (2^g)

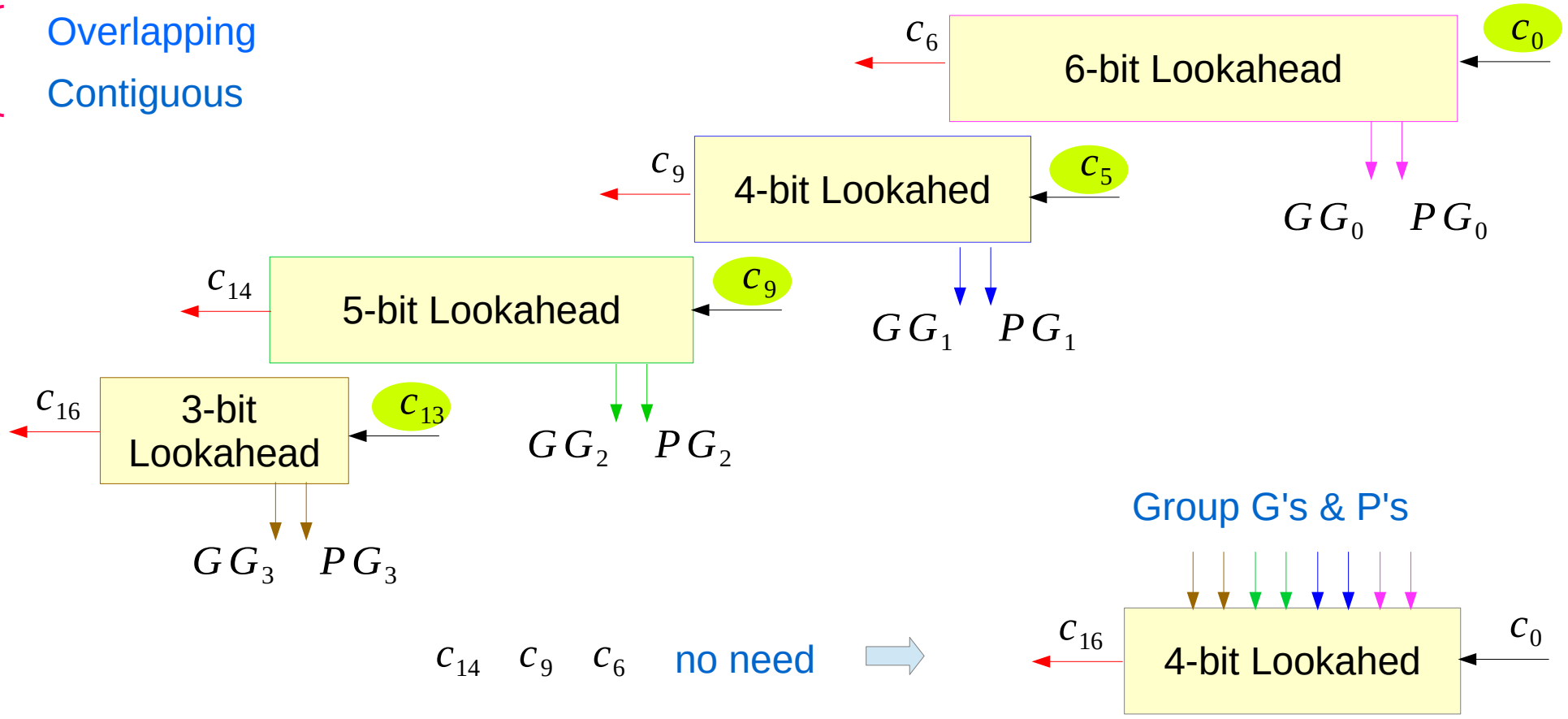
Multi-level Lookahead

Multi-level Carry Lookahead

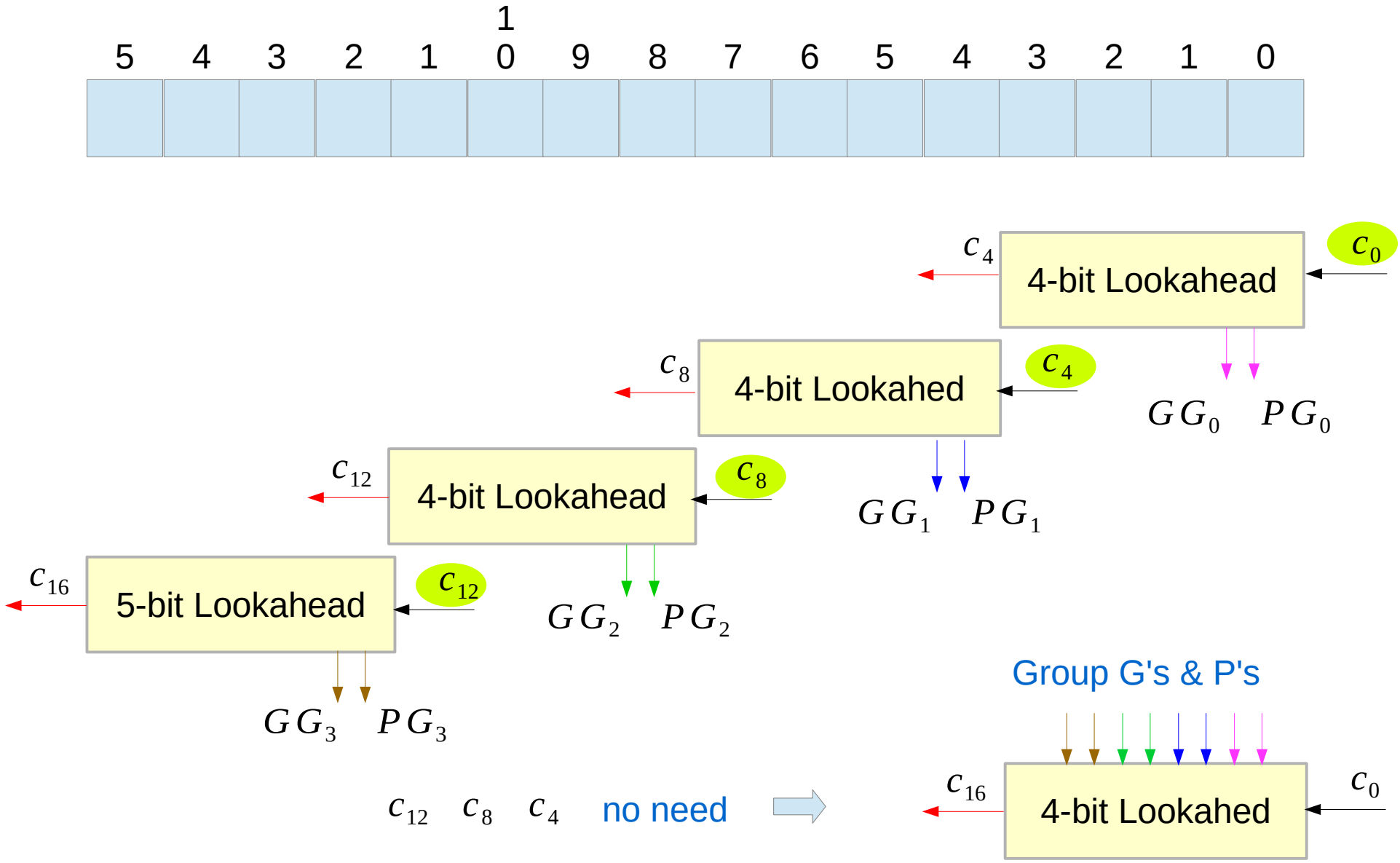


Overlapping

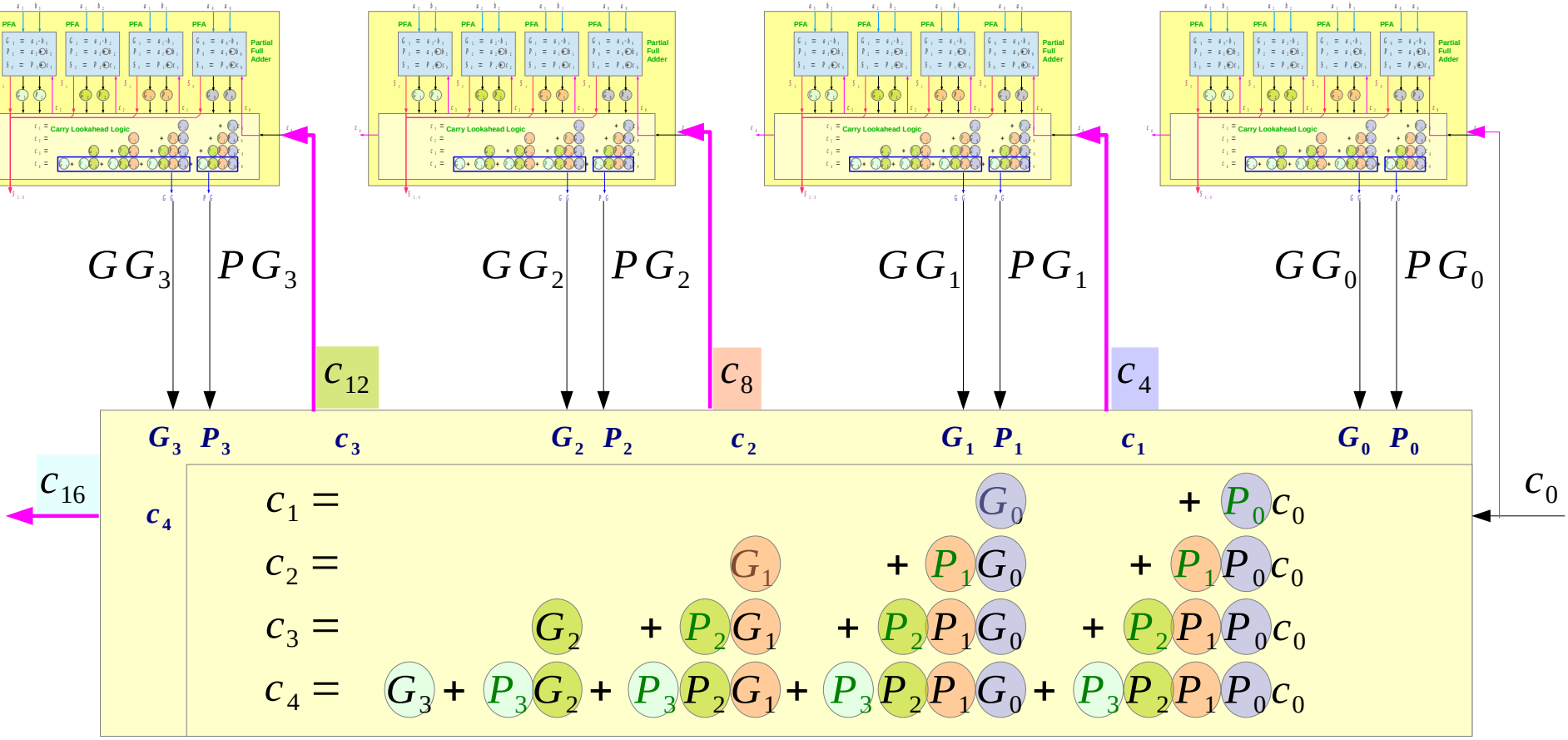
Contiguous



Contiguous Multi-level Carry Lookahead



Group Carry Lookahead Logic



GG and PG (1)

$$c_1 = G_0 + P_0 c_0$$
$$c_2 = G_1 + P_1 [G_0 + P_0 c_0]$$
$$c_3 = G_2 + P_2 [G_1 + P_1 [G_0 + P_0 c_0]]$$
$$c_4 = G_3 + P_3 [G_2 + P_2 [G_1 + P_1 [G_0 + P_0 c_0]]]$$
$$c_4 = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 c_0 = GG_0 + PG_0 c_0$$

$$c_5 = G_4 + P_4 [c_4]$$
$$c_6 = G_5 + P_5 [G_4 + P_4 [c_4]]$$
$$c_7 = G_6 + P_6 [G_5 + P_5 [G_4 + P_4 [c_4]]]$$
$$c_8 = G_7 + P_7 [G_6 + P_6 [G_5 + P_5 [G_4 + P_4 [c_4]]]]$$
$$c_8 = G_7 + P_7 G_6 + P_7 P_6 G_5 + P_7 P_6 P_5 G_4 + P_7 P_6 P_5 P_4 c_4 = GG_1 + PG_1 c_4$$

GG and PG (2)

$$c_9 = G_8 + P_8 c_8$$
$$c_{10} = G_9 + P_9 G_8 + P_9 P_8 c_8$$
$$c_{11} = G_{10} + P_{10} G_9 + P_{10} P_9 G_8 + P_{10} P_9 P_8 c_8$$
$$c_{12} = G_{11} + P_{11} G_{10} + P_{11} P_{10} G_9 + P_{11} P_{10} P_9 G_8 + P_{11} P_{10} P_9 P_8 c_8$$
$$c_{12} = G_{11} + P_{11} G_{10} + P_{11} P_{10} G_9 + P_{11} P_1 P_9 G_8 + P_{11} P_{10} P_9 P_8 c_8 = GG_2 + PG_2 c_8$$

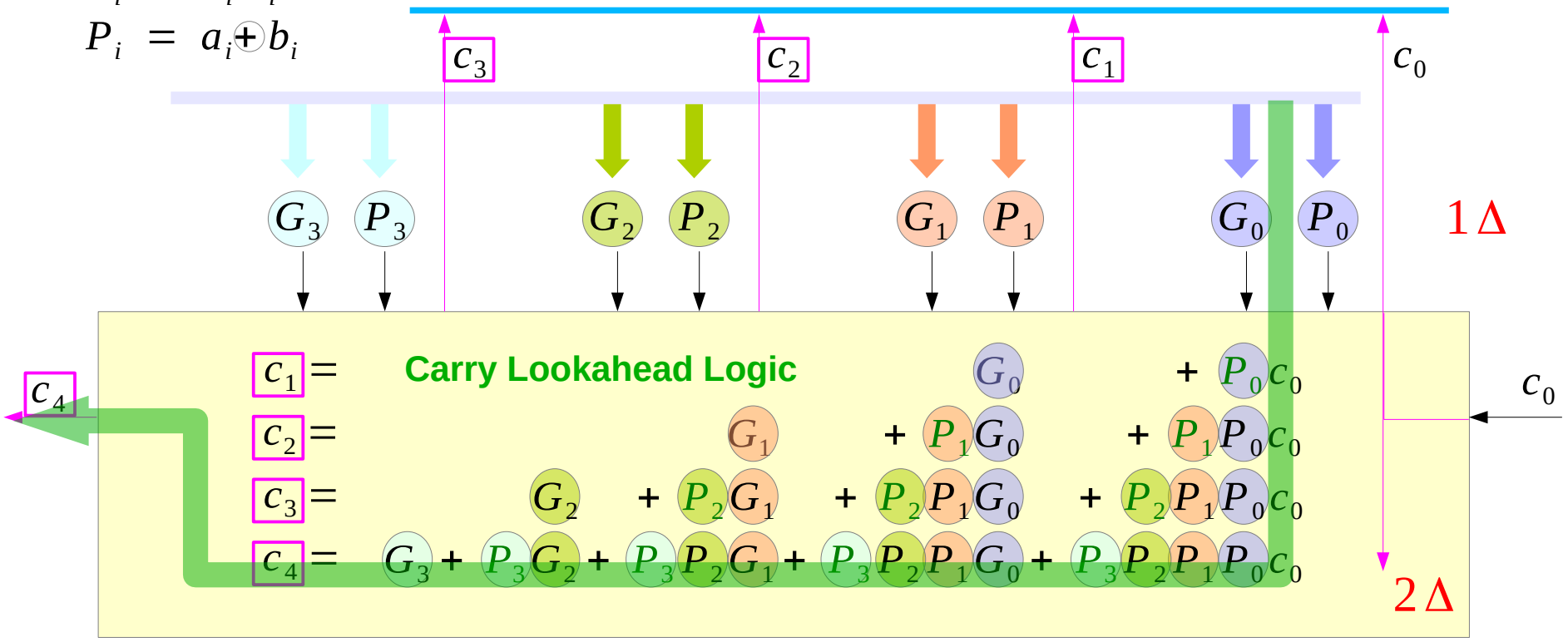
$$c_{13} = G_{12} + P_{12} c_{12}$$
$$c_{14} = G_{13} + P_{13} G_{12} + P_{13} P_{12} c_{12}$$
$$c_{15} = G_{15} + P_{15} G_{13} + P_{15} P_{13} G_{12} + P_{15} P_{13} P_{12} c_{12}$$
$$c_{16} = G_{16} + P_{16} G_{15} + P_{16} P_{15} G_{13} + P_{16} P_{15} P_{13} G_{12} + P_{16} P_{15} P_{13} P_{12} c_{12}$$
$$c_{16} = G_{16} + P_{16} G_{15} + P_{16} P_{15} G_{14} + P_{16} P_{15} P_{14} G_{13} + P_{16} P_{15} P_{14} P_{13} c_0 = GG_3 + PG_3 c_{12}$$

Critical Path in the Carry Lookahead Logic

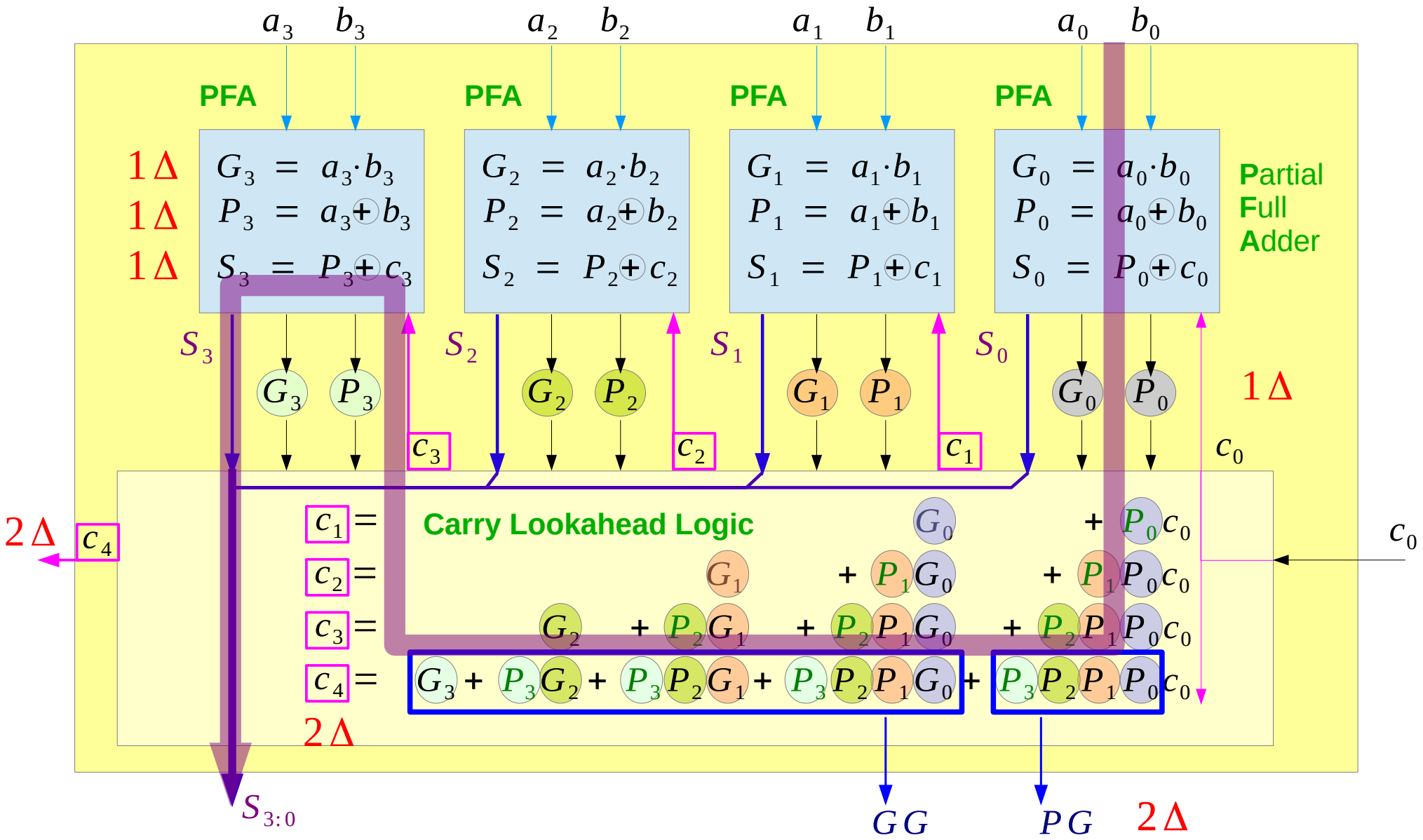
All G_i's P_i's are computed simultaneously from a_i and b_i

each c_i's takes 2 gate delays

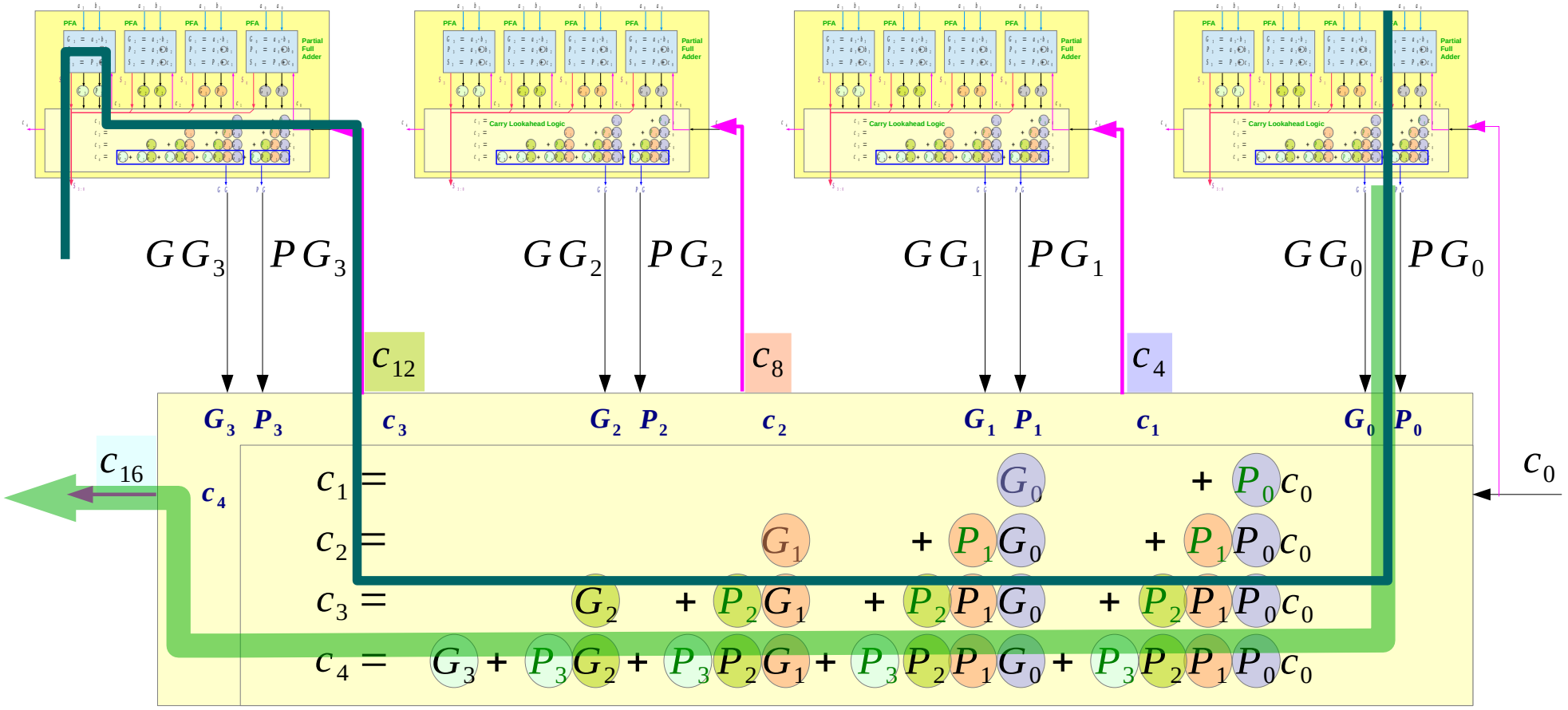
$$G_i = a_i \cdot b_i$$
$$P_i = a_i \oplus b_i$$



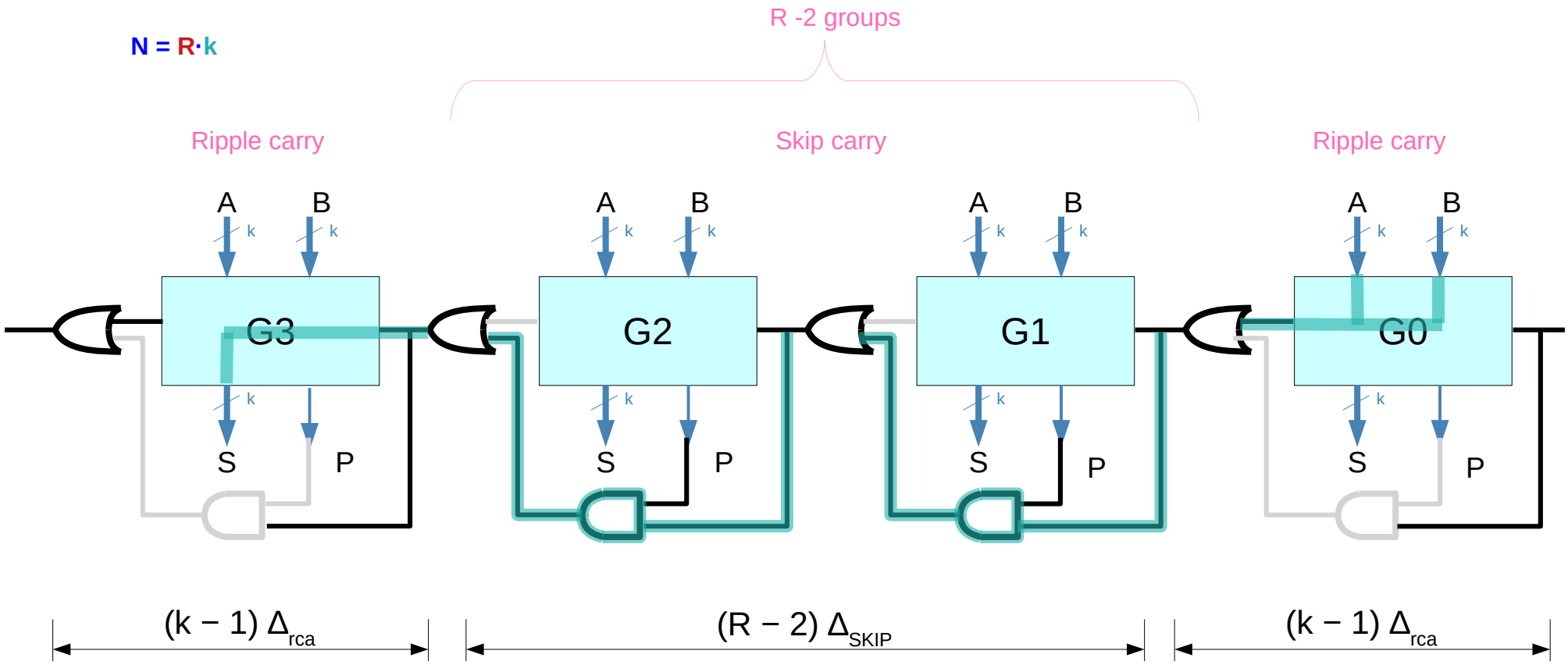
Critical Path in the 4-bit CLA



Critical Path in the Group Carry Lookahead Logic



Carry Skip Adder



Any kill or generate condition results in divided (broken) critical paths

All FA's in R-2 groups must have the propagate condition