

# Carry Lookahead Adder (2A)

---

•  
•

Copyright (c) 2022 - 2013 Young W. Lim.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Please send corrections (or suggestions) to [youngwlim@hotmail.com](mailto:youngwlim@hotmail.com).

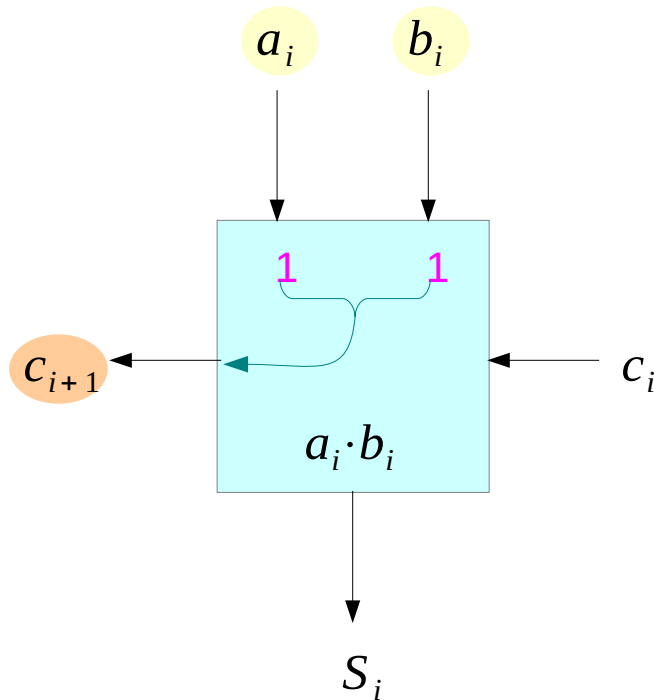
This document was produced by using OpenOffice and Octave.

# G and P

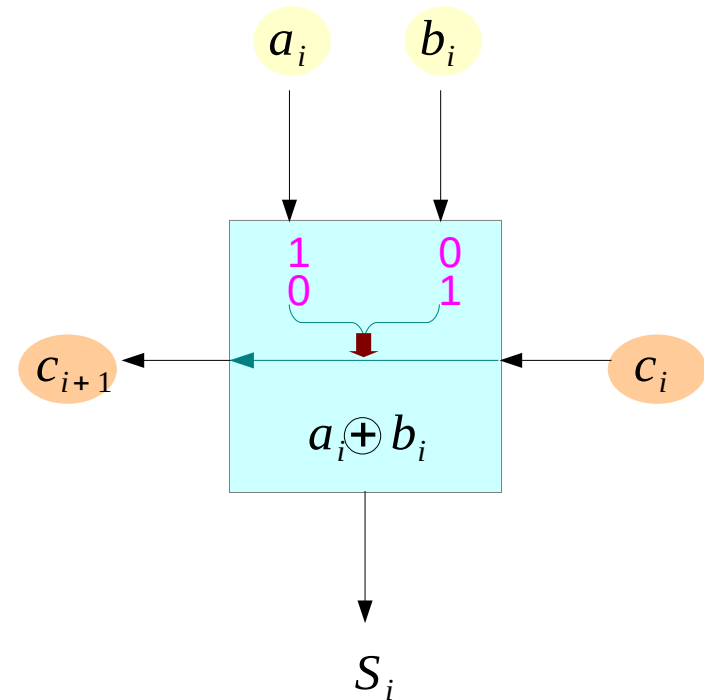
Generate  $G_i = a_i \cdot b_i$

Propagate  $P_i = a_i \oplus b_i$

$$c_{out} = G_i + P_i c_i$$



Generate  $c_{i+1}$



Propagate  $c_i$

# Carry Lookahead Adder

## Carry Lookahead Adder

$$p_i = a_i \oplus b_i$$

$$g_i = a_i \wedge b_i$$

$$c_1 = g_0 + p_0 \wedge c_0$$

$$c_2 = g_1 + p_1 \wedge c_1$$

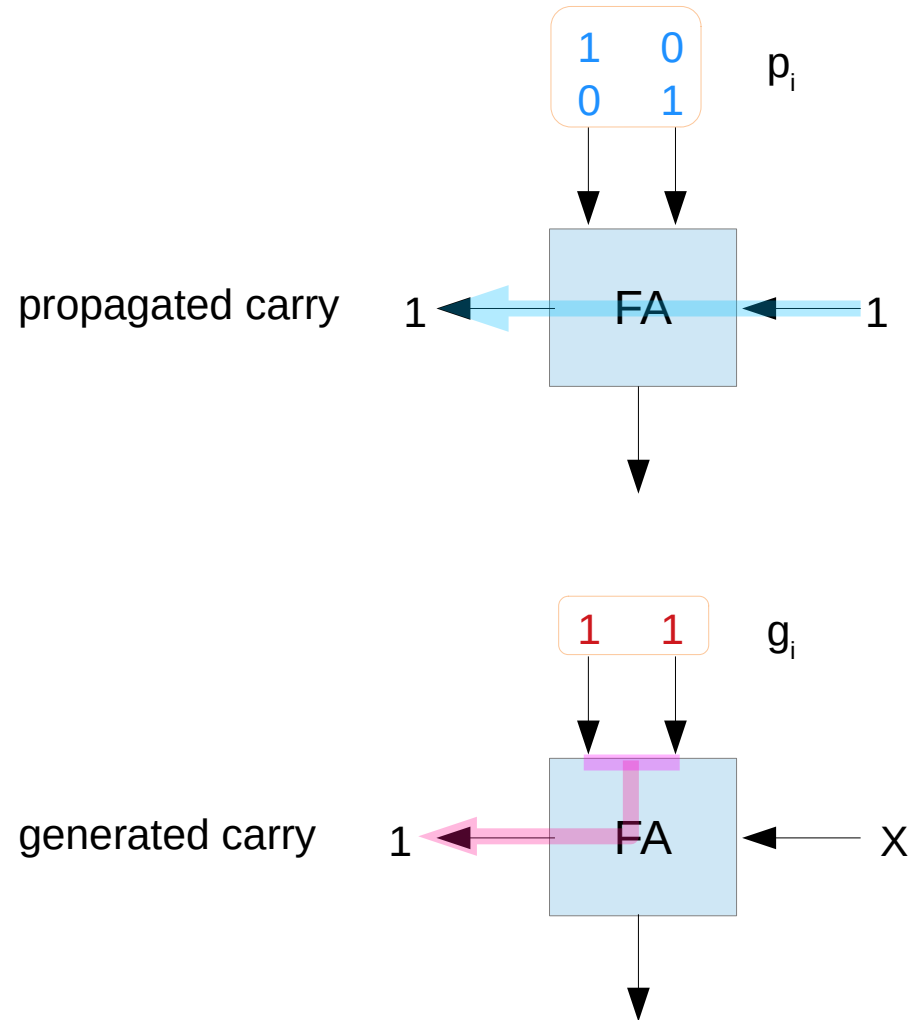
$$c_3 = g_2 + p_2 \wedge c_2$$

$$c_4 = g_3 + p_3 \wedge c_3$$

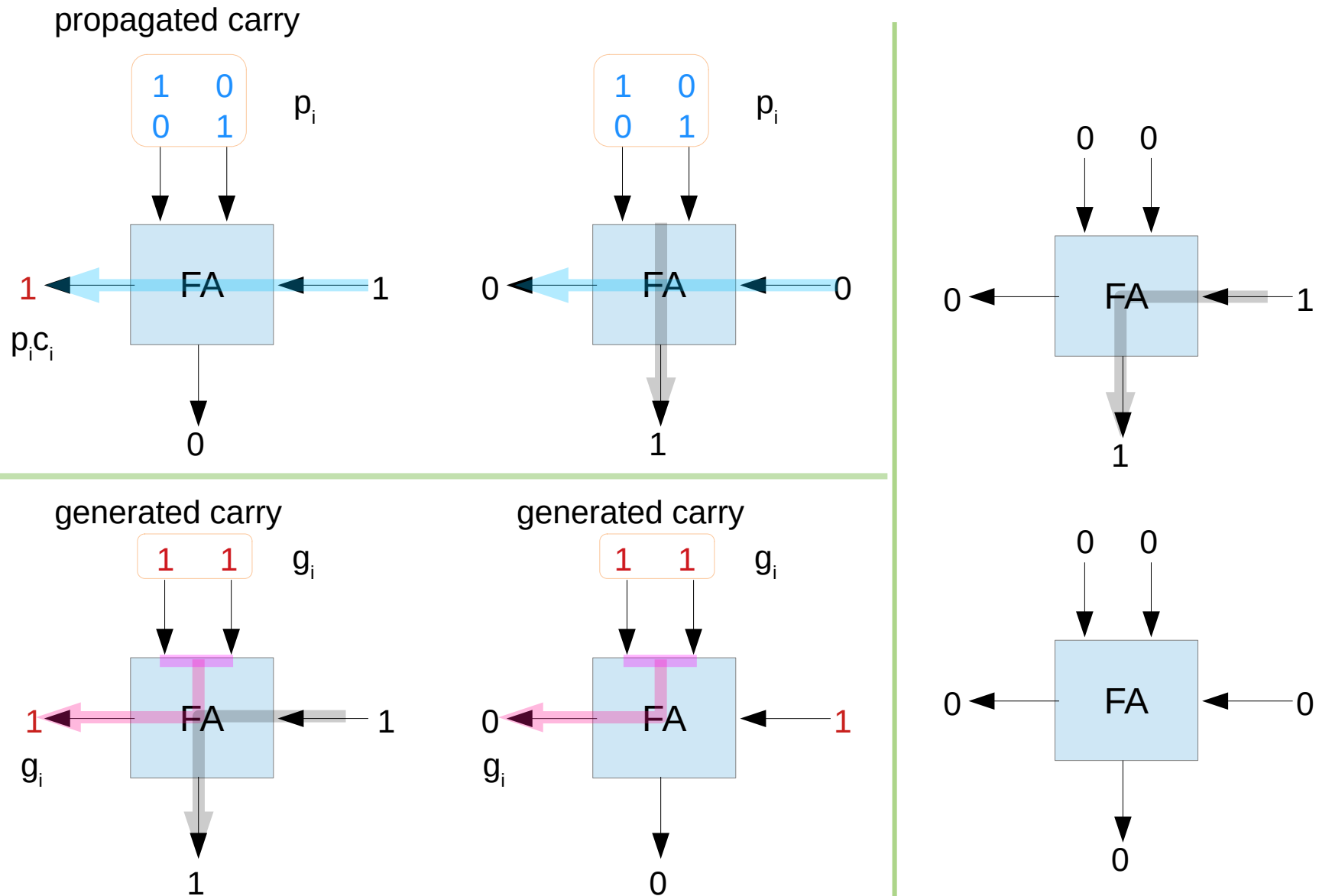
generated carry

propagated carry

[https://en.wikipedia.org/wiki/Carry-skip\\_adder](https://en.wikipedia.org/wiki/Carry-skip_adder)



# Propagated and Generated Carries



# Invert

Invert

$$(C1_3 \oplus C1_2)$$

P P  
G G

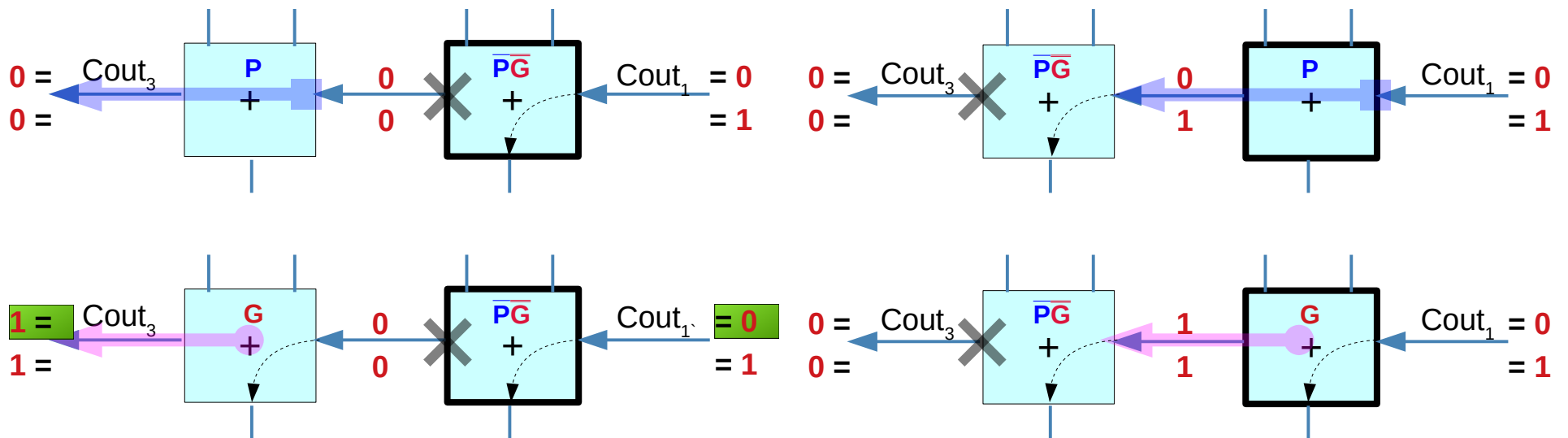
$$C1 = X + Y$$

$$C0 = X \cdot Y$$

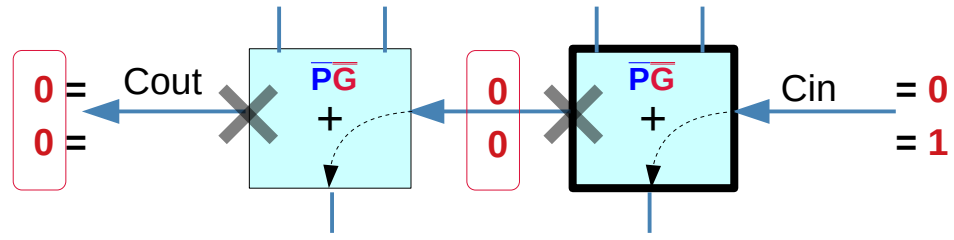
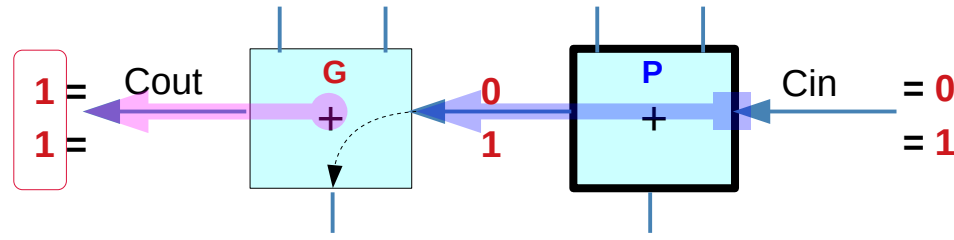
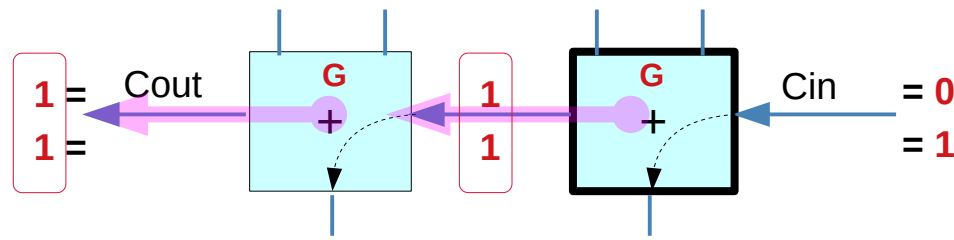
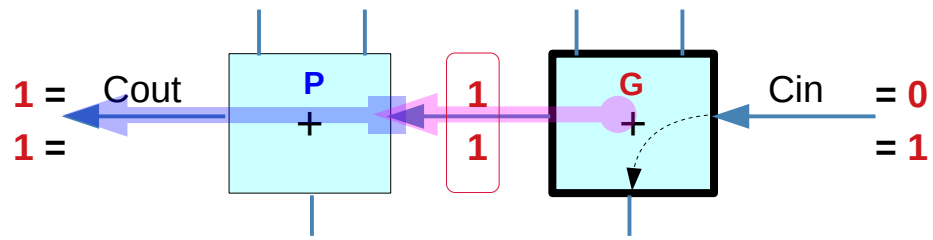
$$C1 \oplus C0 = X \oplus Y = P$$

P G  
G

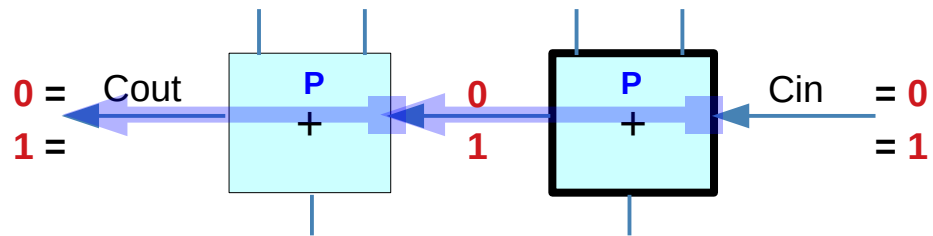
P	P'G'
G	P'G'
P'G'	P
P'G'	G



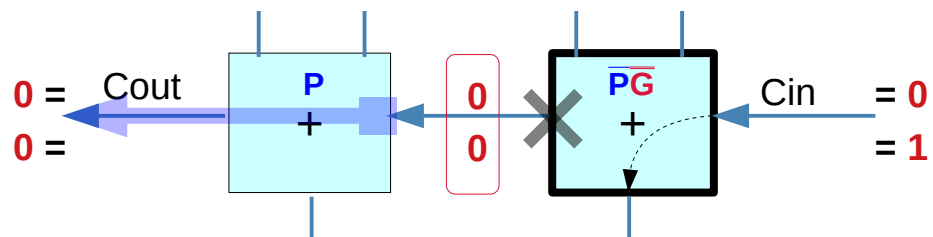
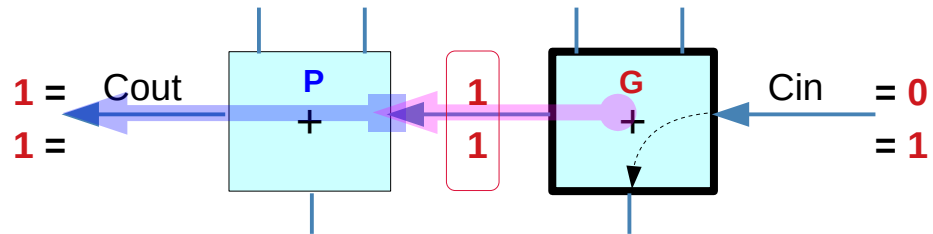
# Variable Block



# Variable Block



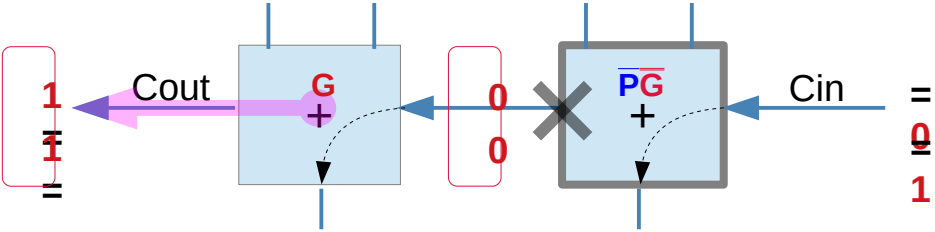
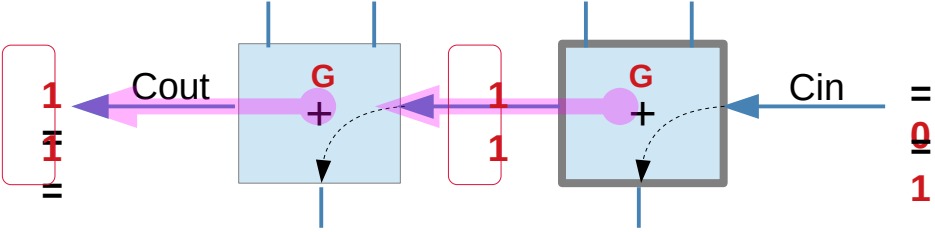
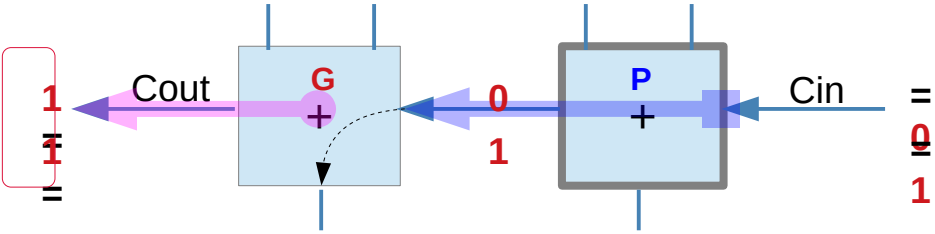
Propagate



Invert

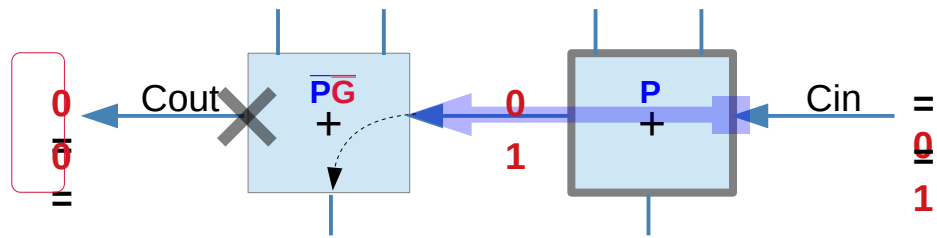


# Variable Block

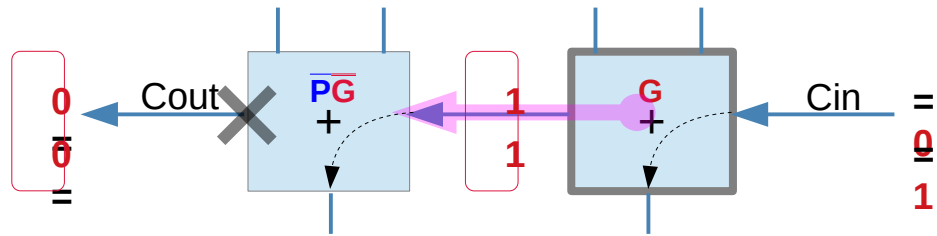


Invert

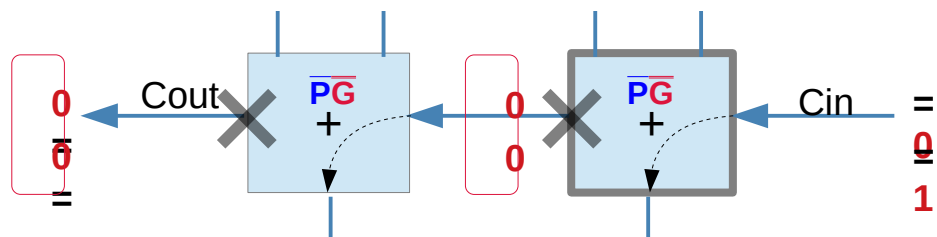
# Variable Block



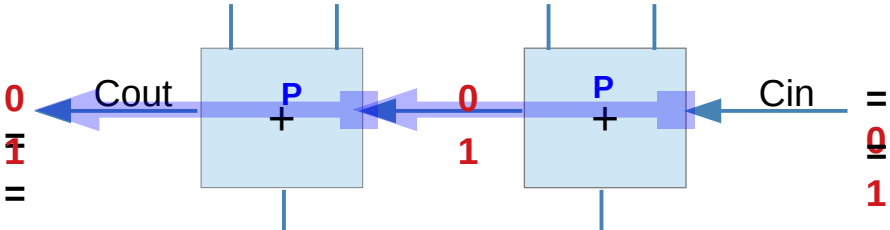
Invert



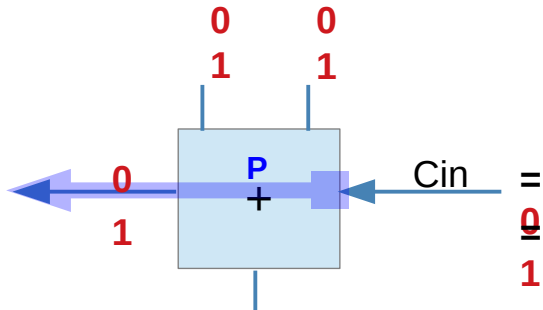
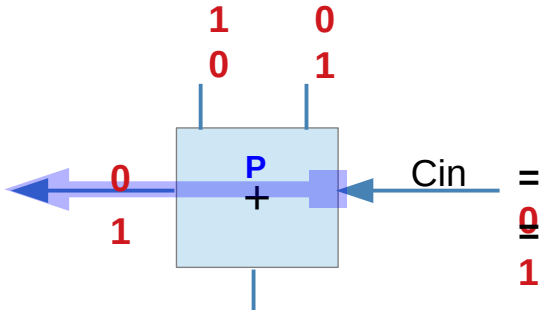
Invert



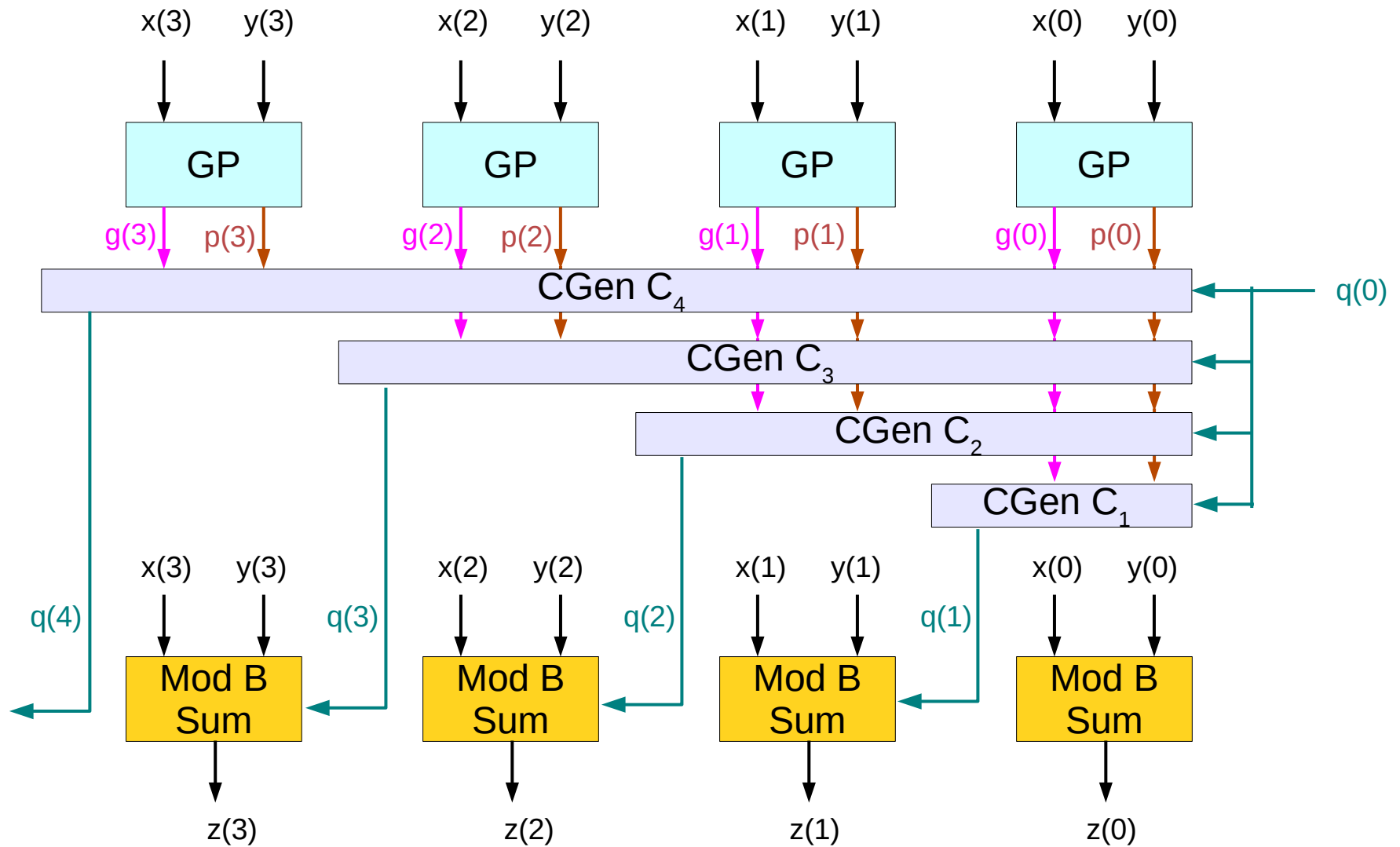
# Variable Block



Propagate



# Carry Lookahead Adder



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

# Carry Equations

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

$$c_1 = G_0 + P_0 c_0$$

$$c_2 = G_1 + P_1 c_1$$

$$c_3 = G_2 + P_2 c_2$$

$$c_4 = G_3 + P_3 c_3$$

$$c_1 = G_0 + P_0 c_0$$

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

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

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

$$G_0 + P_0 c_0 = c_1$$

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

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

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

# Carry Equations into Gates (1)

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

$$c_1 = G_0 + P_0 c_0$$

$$c_2 = G_1 + P_1 c_1$$

$$c_3 = G_2 + P_2 c_2$$

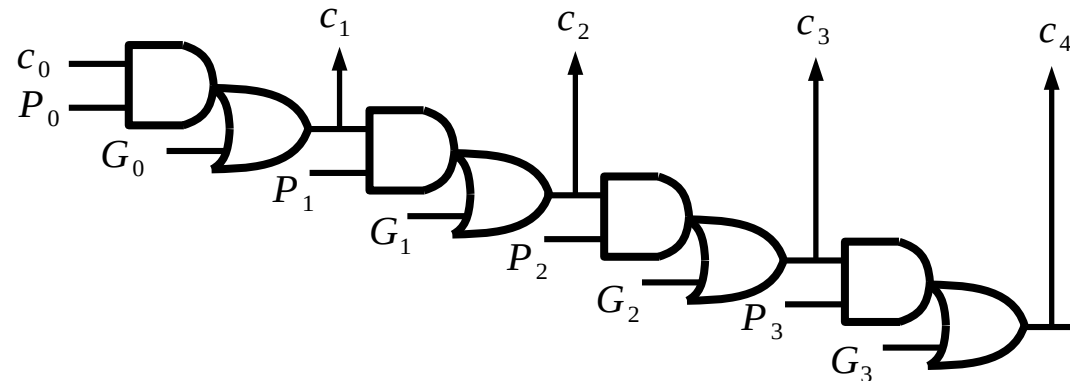
$$c_4 = G_3 + P_3 c_3$$

$$c_1 = G_0 + P_0 c_0$$

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

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

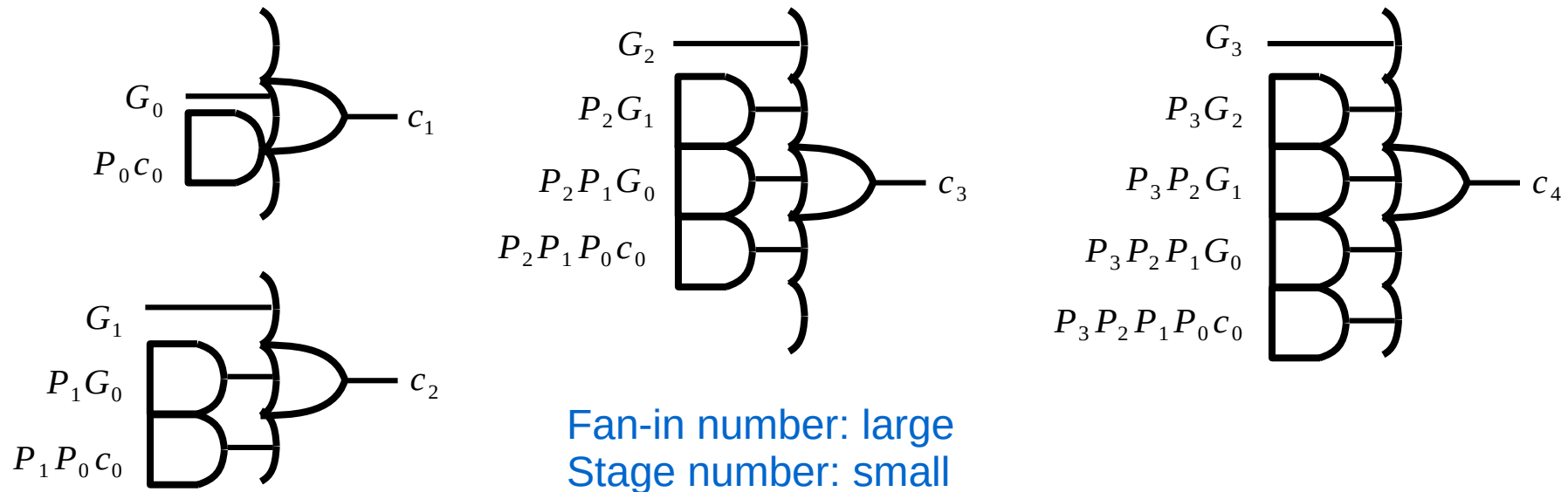
$$c_4 = G_3 + P_3 [G_2 + P_2 [G_1 + P_1 [G_0 + P_0 c_0]]]$$



Fan-in number: small  
Stage number: large

# Carry Equations into Gates (2)

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



# 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

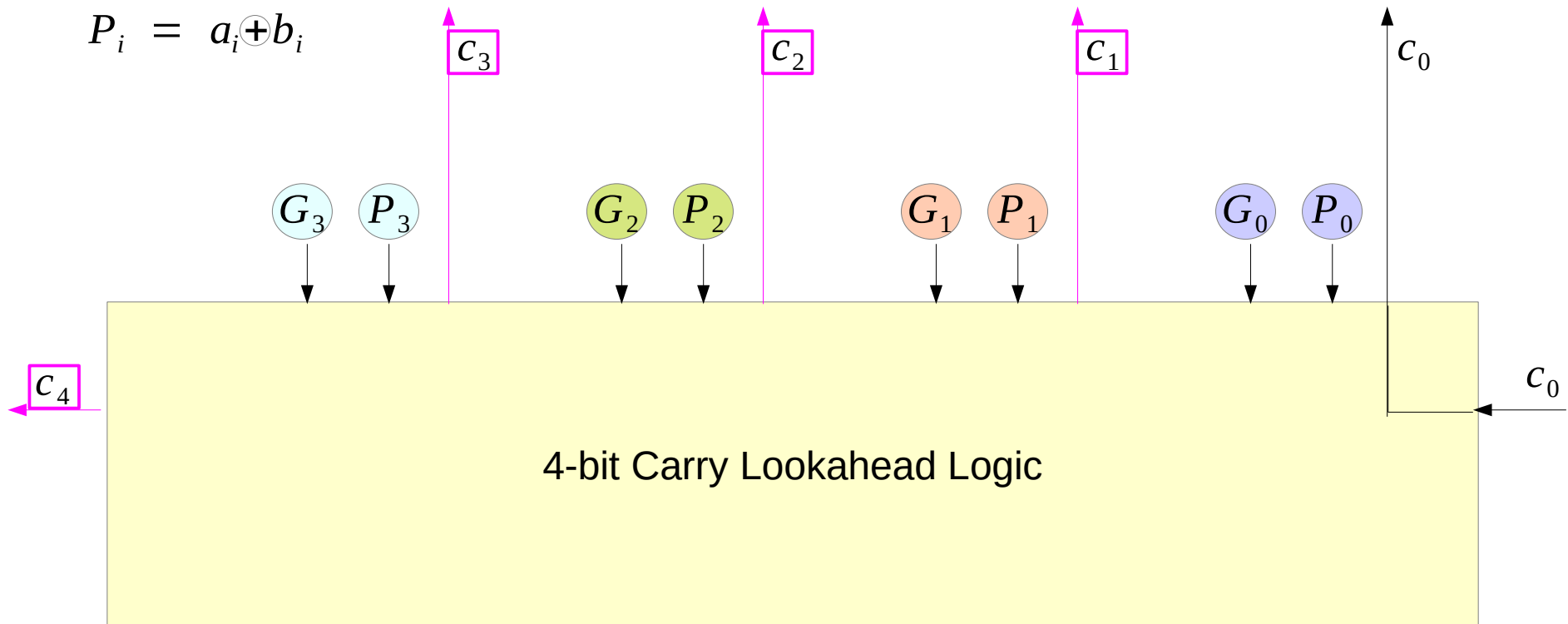


# 4-bit Carry Lookahead Logic – interface

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

each  $c_i$ 's takes 2 gate delays

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



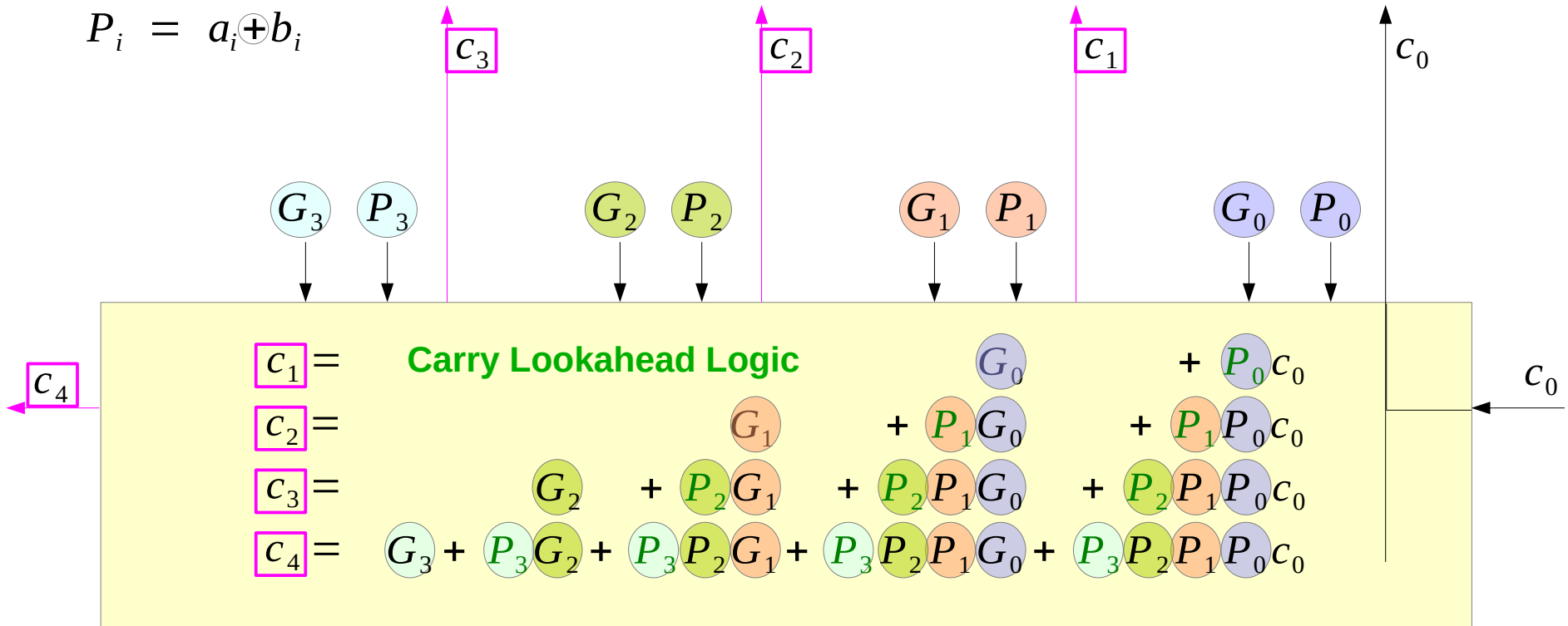
# 4-bit Carry Lookahead Logic – boolean equations

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

each  $c_i$ 's takes 2 gate delays

$$G_i = a_i \cdot b_i$$

$$P_i = a_i \oplus b_i$$



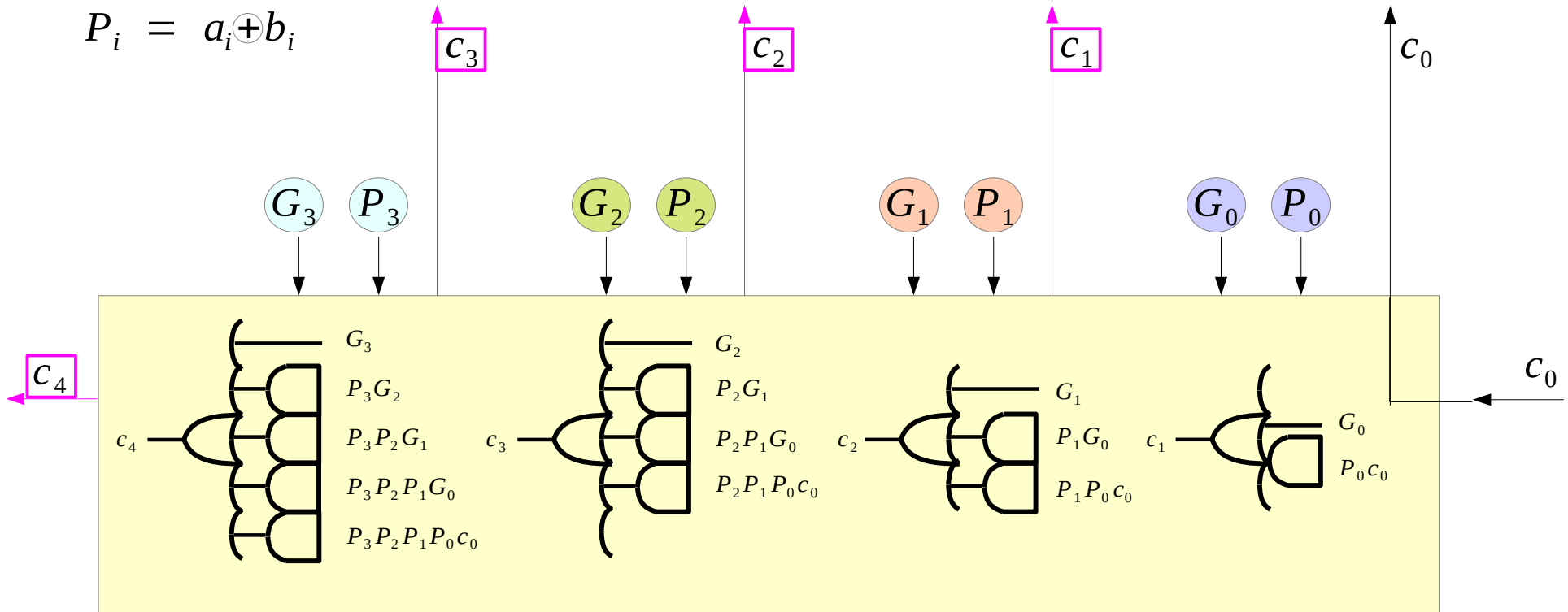
# 4-bit Carry Lookahead Logic – POS

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

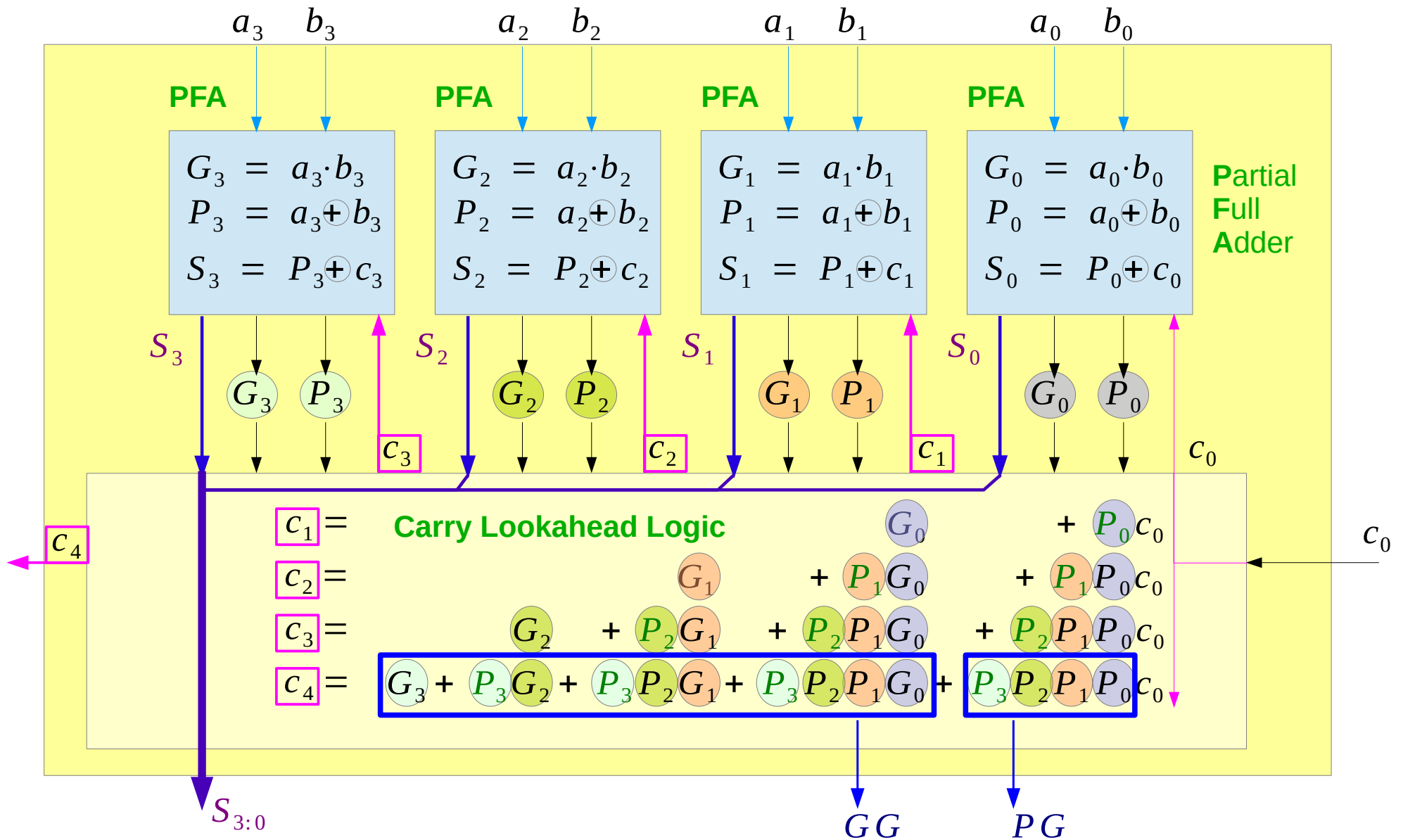
each  $c_i$ 's takes 2 gate delays

$$G_i = a_i \cdot b_i$$

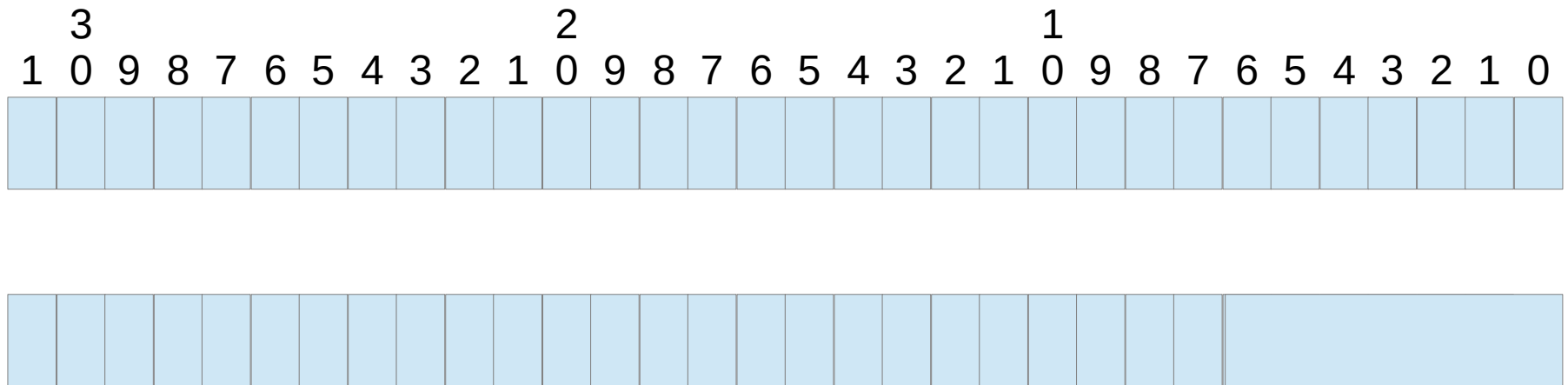
$$P_i = a_i \oplus b_i$$



# 4-bit CLA

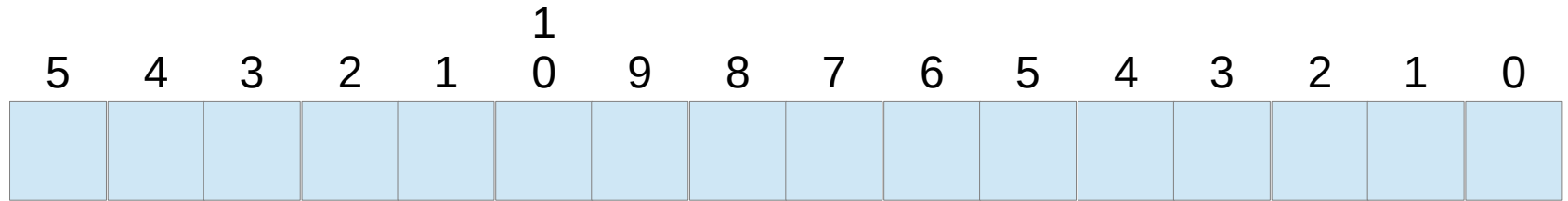


# Multi-level Carry Lookahead

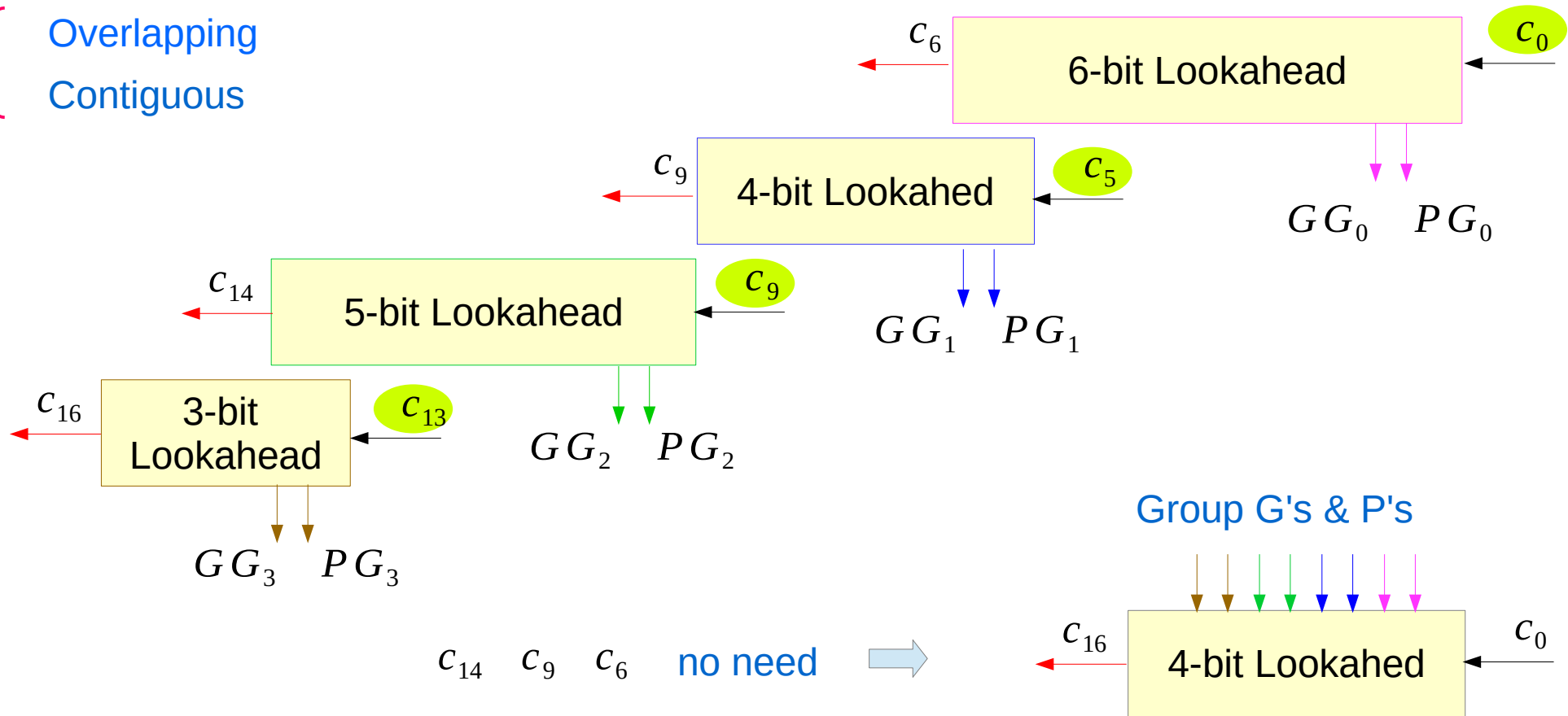


{ High Radix Addition ( $2^g$ )  
Multi-level Lookahead

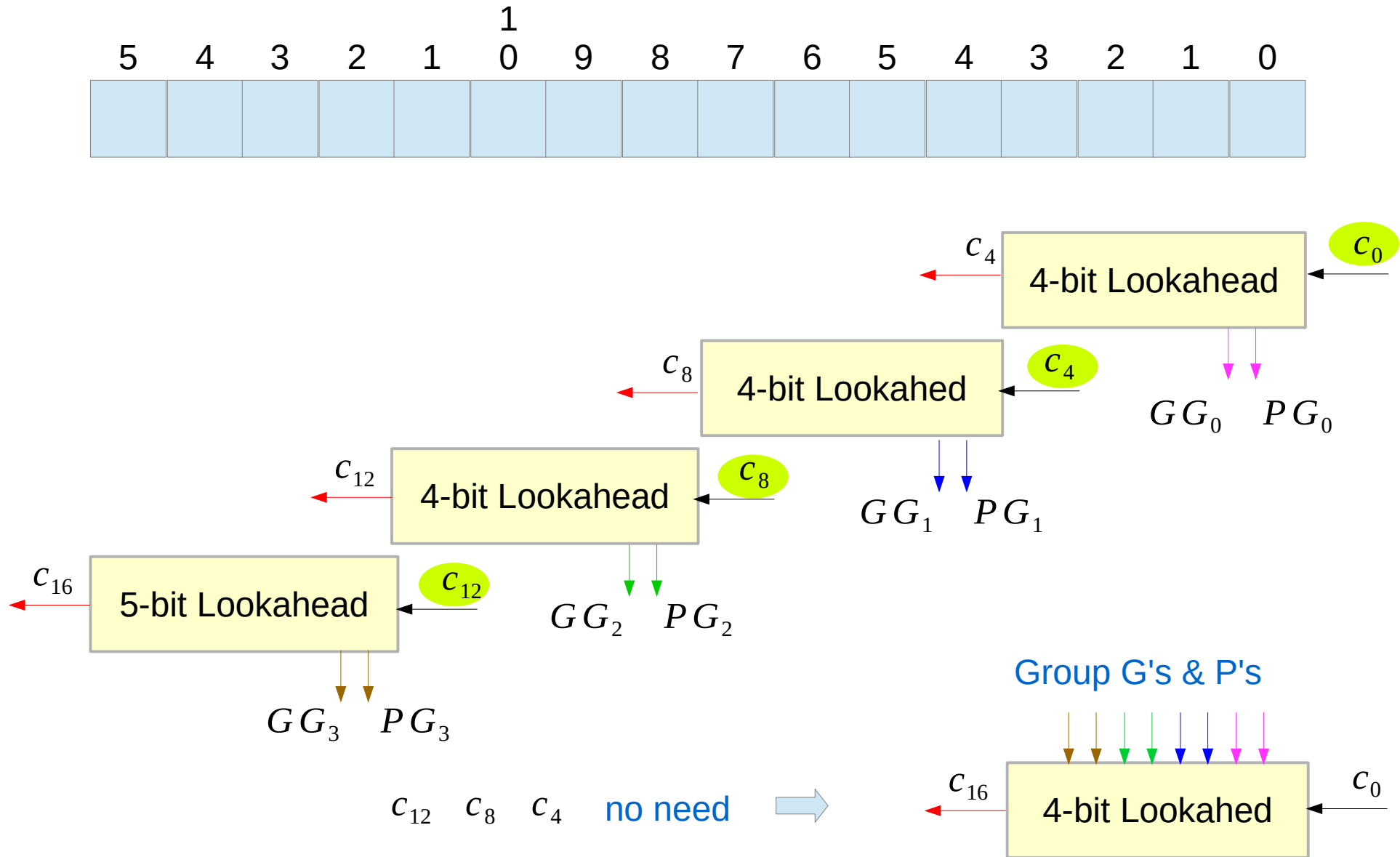
# Multi-level Carry Lookahead



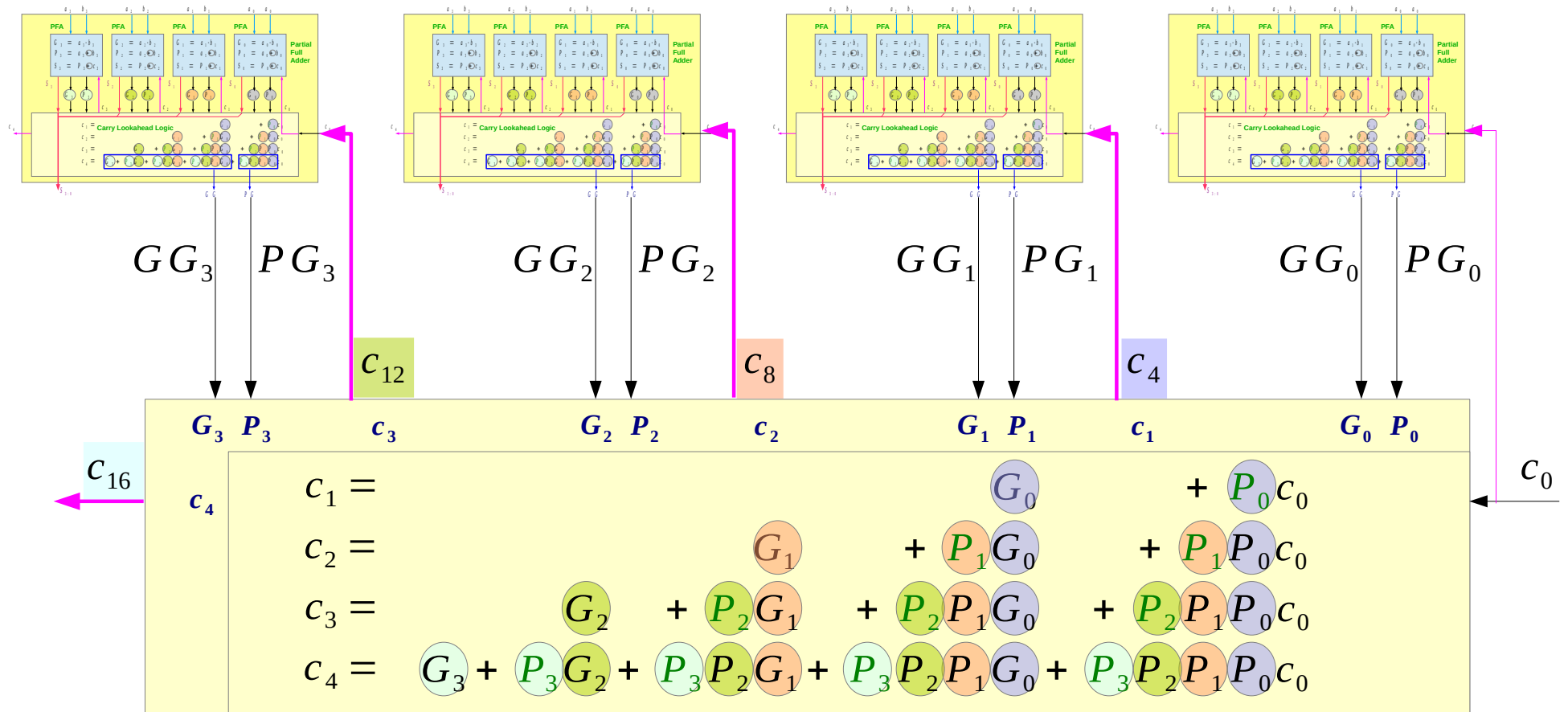
Overlapping  
Contiguous



# 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)

$$c_9 = G_8 + P_8 [c_8]$$

$$c_{10} = G_9 + P_9 [G_8 + P_8 [c_8]]$$

$$c_{11} = G_{10} + P_{10} [G_9 + P_9 [G_8 + P_8 [c_8]]]$$

$$c_{12} = G_{11} + P_{11} [G_{10} + P_{10} [G_9 + P_9 [G_8 + P_8 [c_8]]]]$$

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

$$c_{13} = G_{12} + P_{12} [c_{12}]$$

$$c_{14} = G_{13} + P_{13} [G_{12} + P_{12} [c_{12}]]$$

$$c_{15} = G_{15} + P_{15} [G_{13} + P_{13} [G_{12} + P_{12} [c_{12}]]]$$

$$c_{16} = G_{16} + P_{16} [G_{15} + P_{15} [G_{13} + P_{13} [G_{12} + P_{12} [c_{12}]]]]$$

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

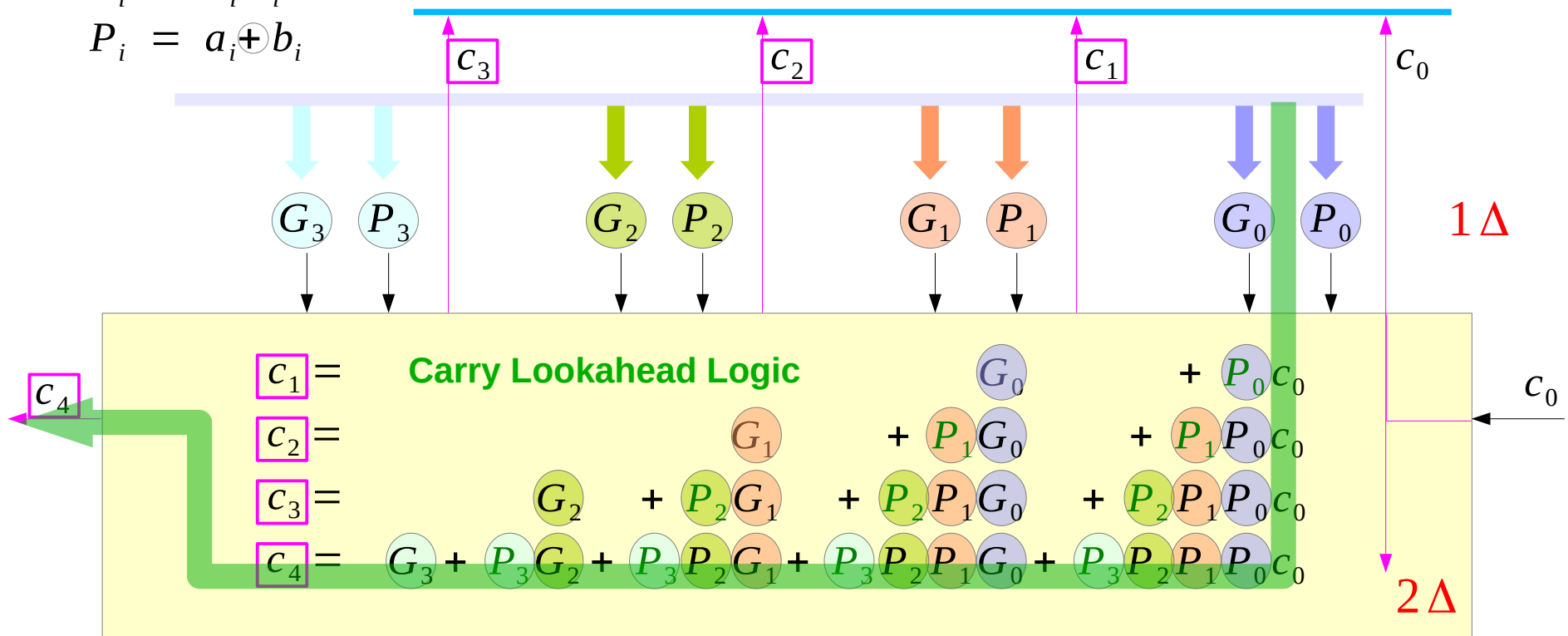
# Critical Path in the Carry Lookahead Logic

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

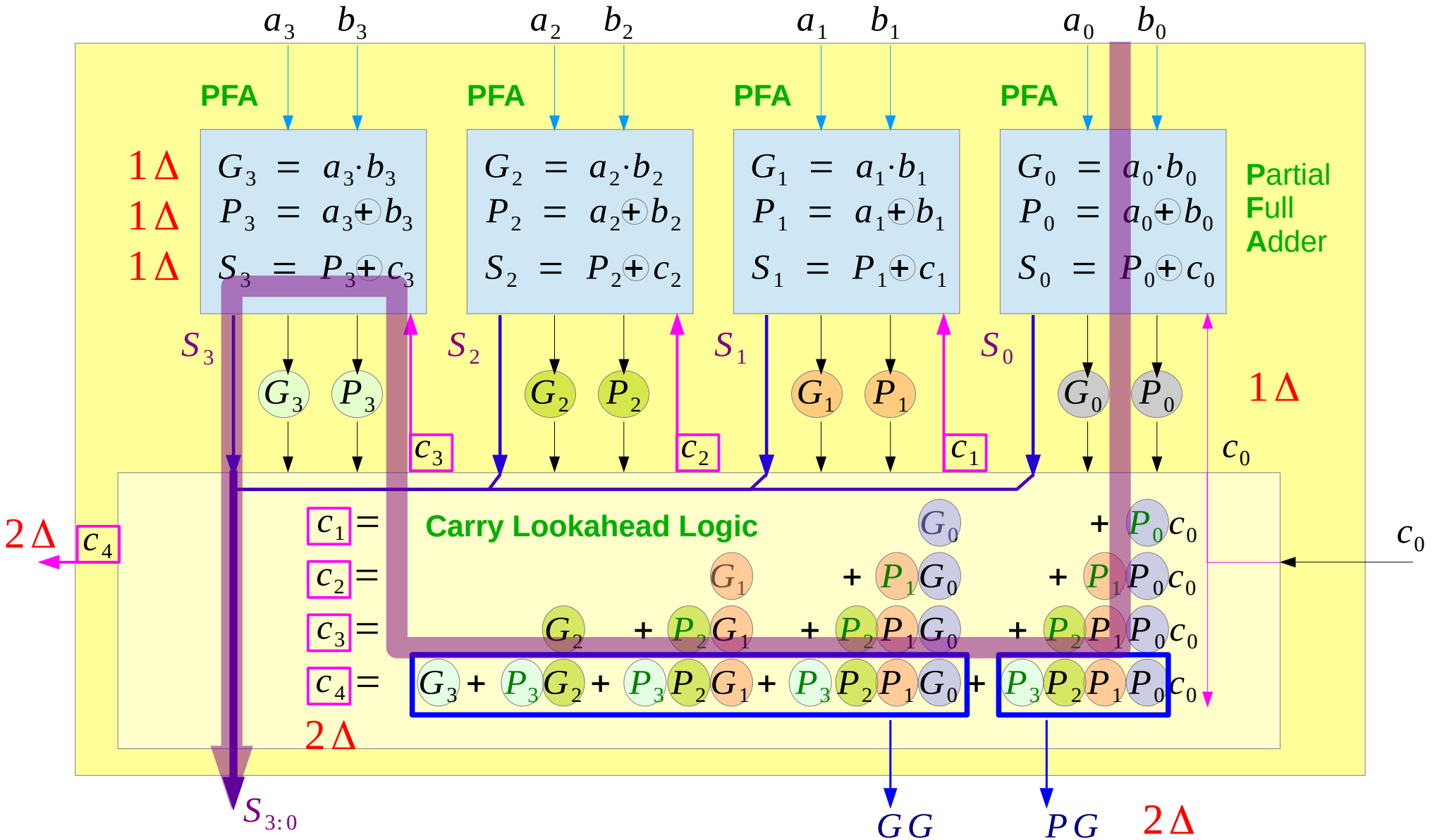
each  $c_i$ 's takes 2 gate delays

$$G_i = a_i \cdot b_i$$

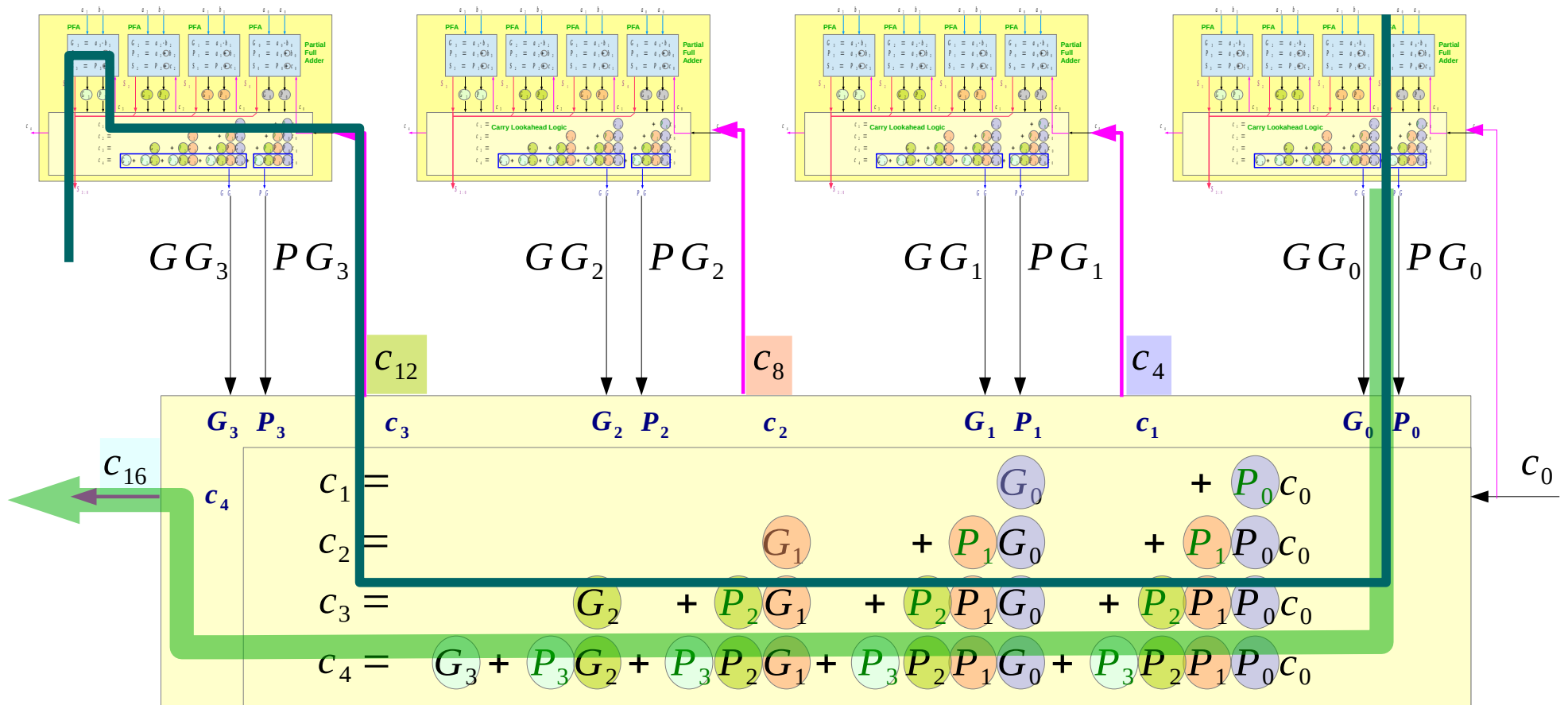
$$P_i = a_i \oplus b_i$$



# Critical Path in the 4-bit CLA



# Critical Path in the Group Carry Lookahead Logic



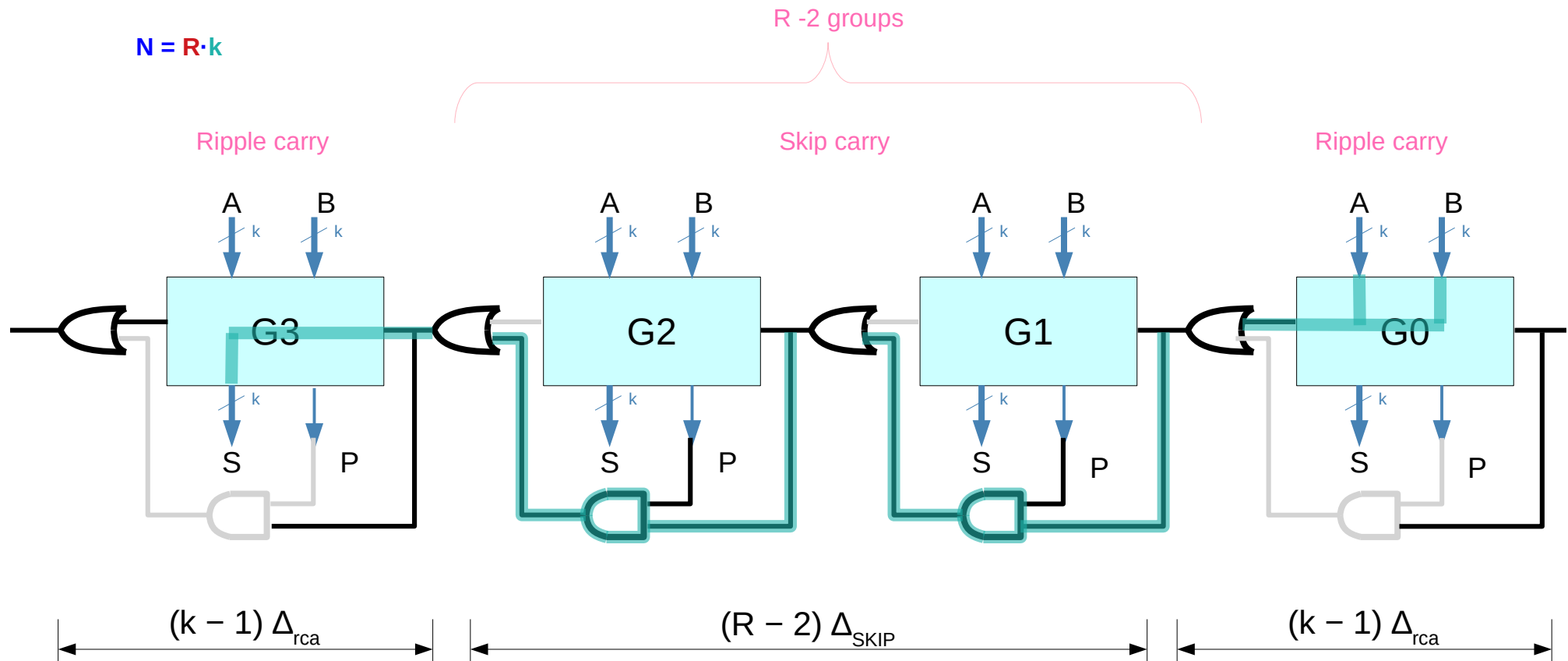








# Carry Skip Adder



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

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