Carry Lookahead Adder (2A)

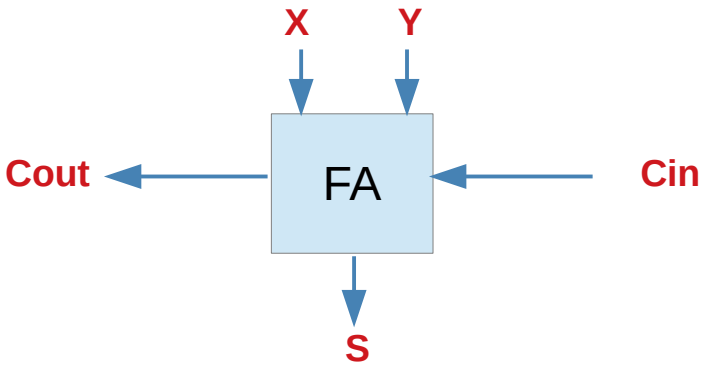
-
-

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

This document was produced by using OpenOffice and Octave.

Carry Kill, Propagate, Generate conditions (1)

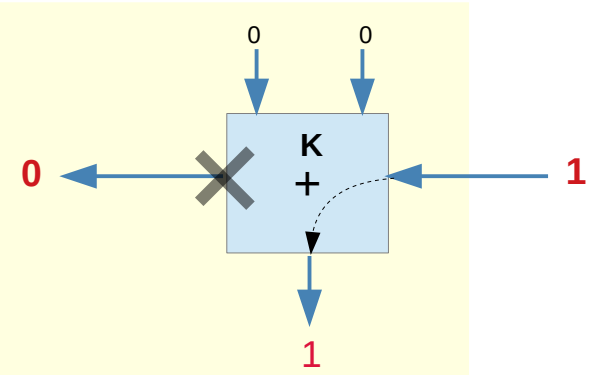
X	Y		
0	0	K	Kill (=$\overline{P}\overline{G}$)
0	1	P	Propagate
1	0	P	Propagate
1	1	G	Generate



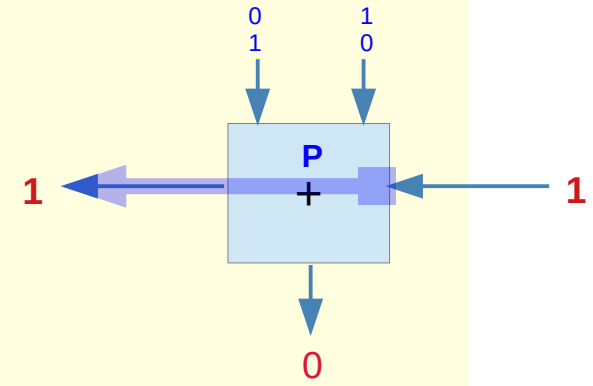
https::/electronics.stackexchange.com/questions/21251/critical-path-for-carry-skip-adder

Carry Kill, Propagate, Generate conditions (2)

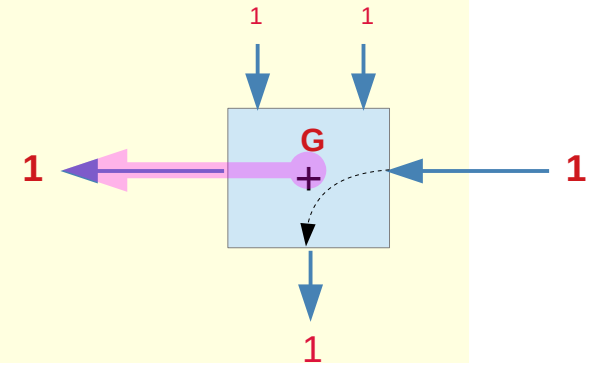
killed carry



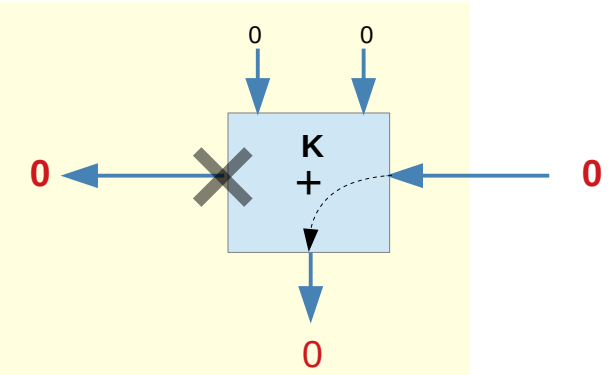
propagated carry



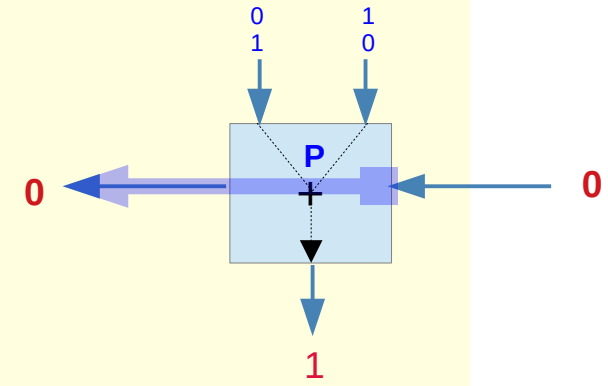
generated carry



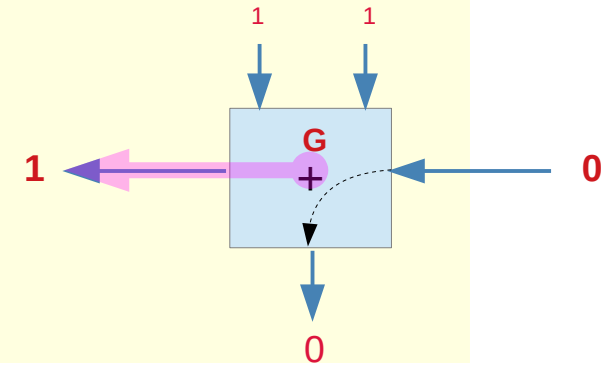
killed carry



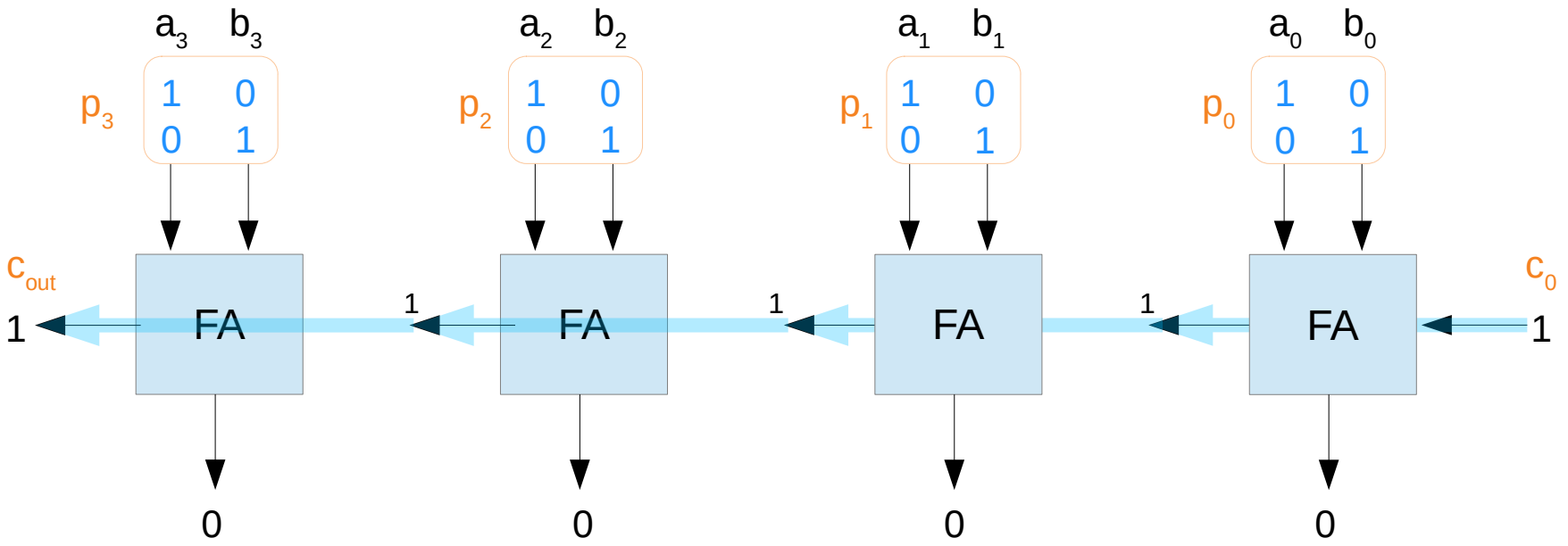
propagated carry



generated carry



C_0 propagation condition

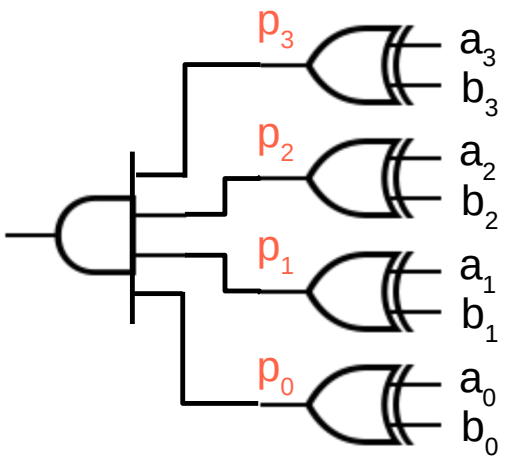


$s = p_3 \wedge p_2 \wedge p_1 \wedge p_0$

$= (a_3 \oplus b_3) \wedge (a_2 \oplus b_2) \wedge (a_1 \oplus b_1) \wedge (a_0 \oplus b_0)$

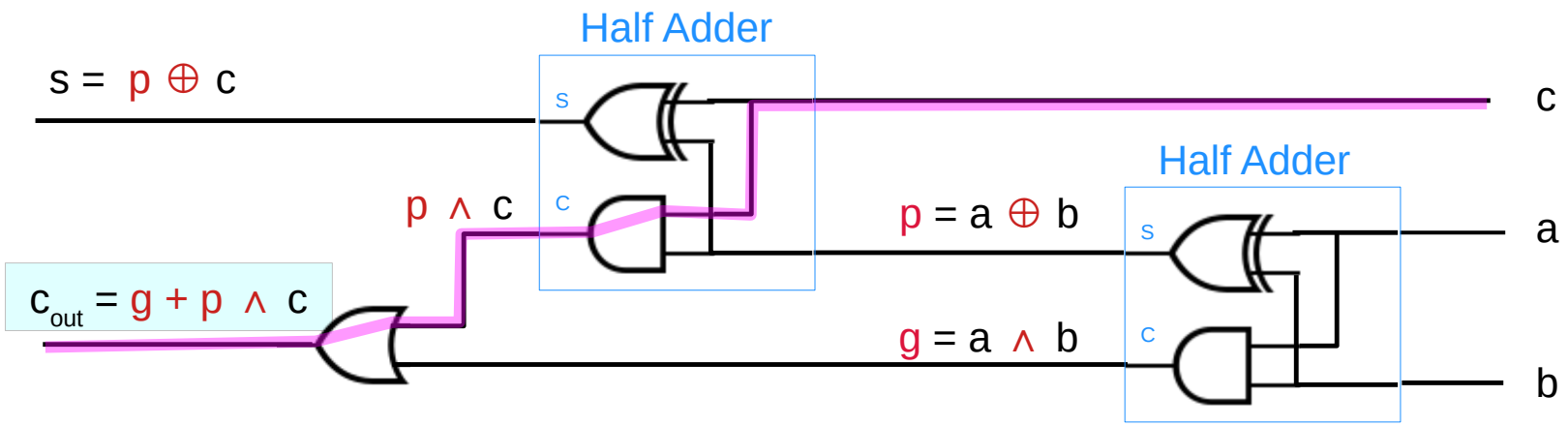
c_0 can be propagated to c_{out}
only if $s = 1$

$s = p_3 p_2 p_1 p_0$



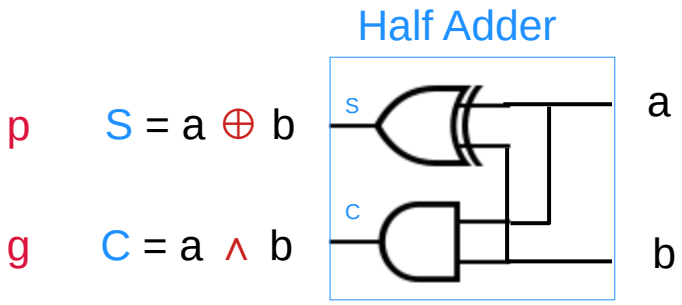
https://en.wikipedia.org/wiki/Carry-skip_adder

A FA in terms of **p** and **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.

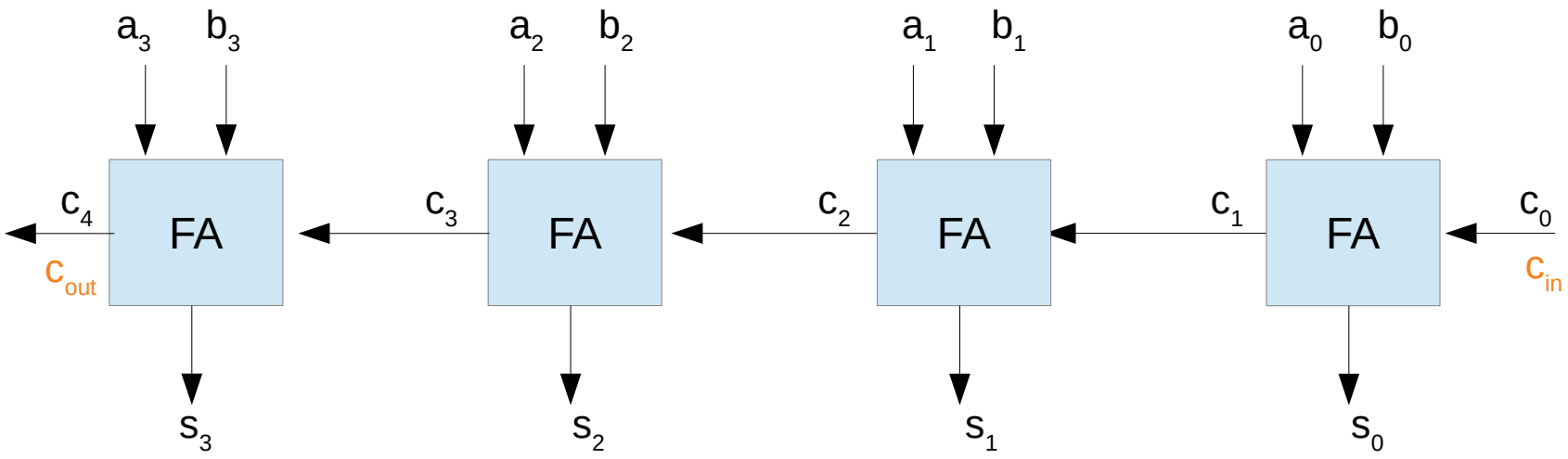
Carry Equations in a Ripple Carry Adder

$$c_1 = g_0 + p_0 \wedge c_0$$
$$c_2 = g_1 + p_1 \wedge c_1$$
$$c_3 = g_2 + p_2 \wedge c_2$$
$$c_4 = g_3 + p_3 \wedge c_3$$

generated carry

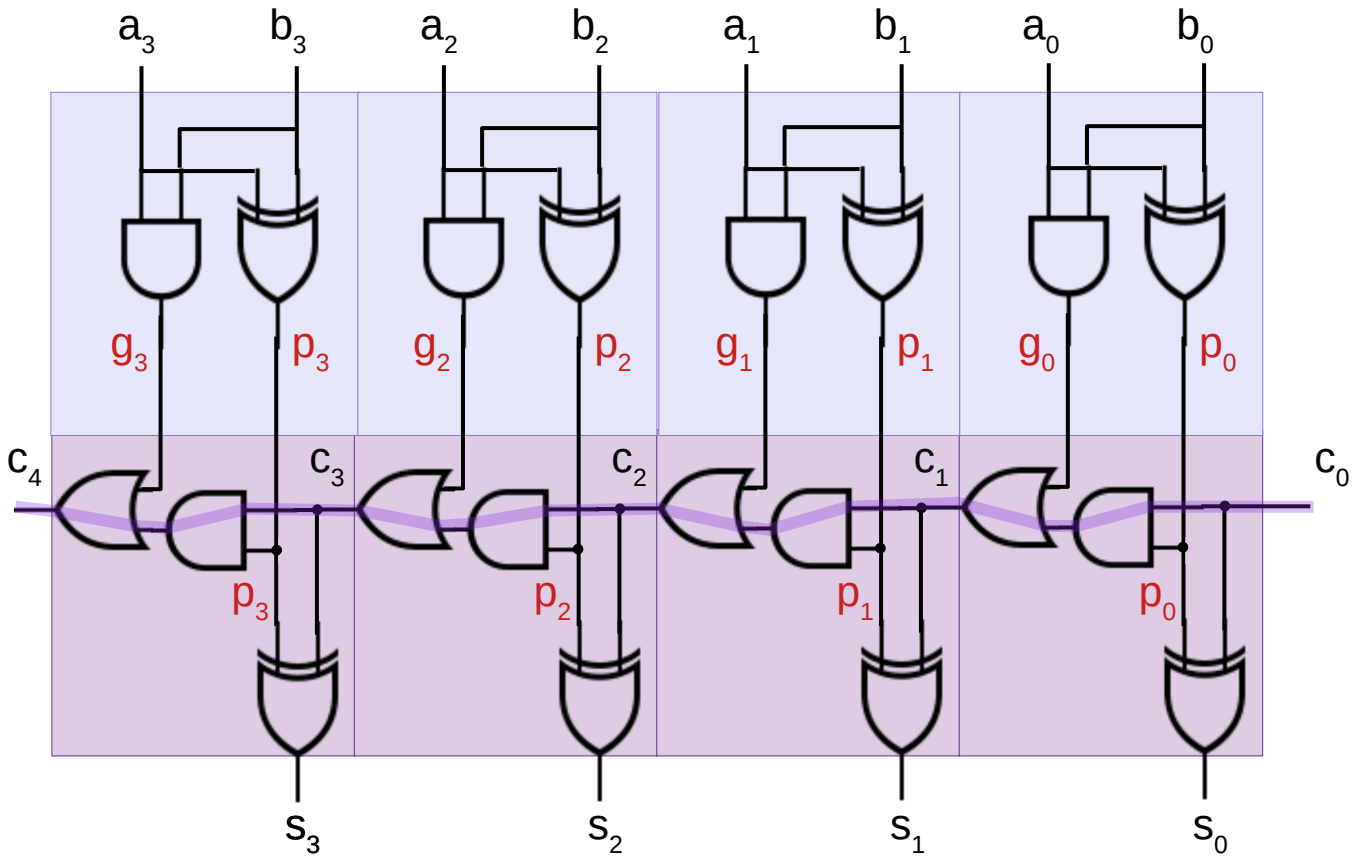
propagated carry

$$p_i = a_i \oplus b_i$$
$$g_i = a_i \wedge b_i$$



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

A 4-bit Ripple Carry Adder



Half Adder 1

$p_i = a_i \oplus b_i$

$g_i = a_i \wedge b_i$

Half Adder 2

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

$s_i = p_i \oplus c_i$

Critical Path : $c_4 - c_3 - c_2 - c_1 - c_0$: 8 gate delay

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

Expanding carry equations

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

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

$$c_2 = g_1 + p_1 c_1 \rightarrow c_2 = g_1 + p_1 \left[g_0 + p_0 c_0 \right]$$

$$c_3 = g_2 + p_2 c_2 \rightarrow c_3 = g_2 + p_2 \left[g_1 + p_1 \left[g_0 + p_0 c_0 \right] \right]$$

$$c_4 = g_3 + p_3 c_3 \rightarrow c_4 = g_3 + p_3 \left[g_2 + p_2 \left[g_1 + p_1 \left[g_0 + p_0 c_0 \right] \right] \right]$$

c_{out} c_{in}

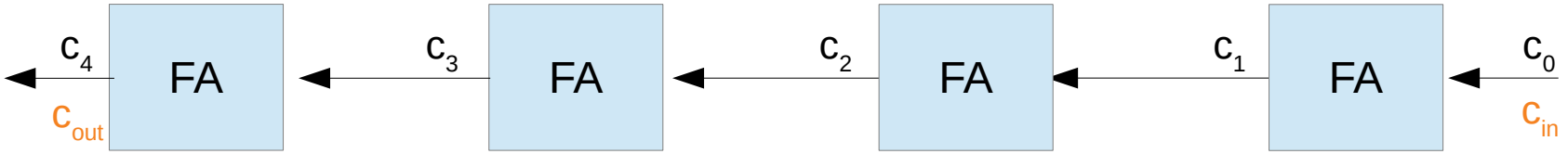
$$g_0 + p_0 c_0 = c_1$$

$$g_1 + p_1 g_0 + p_1 p_0 c_0 = c_2$$

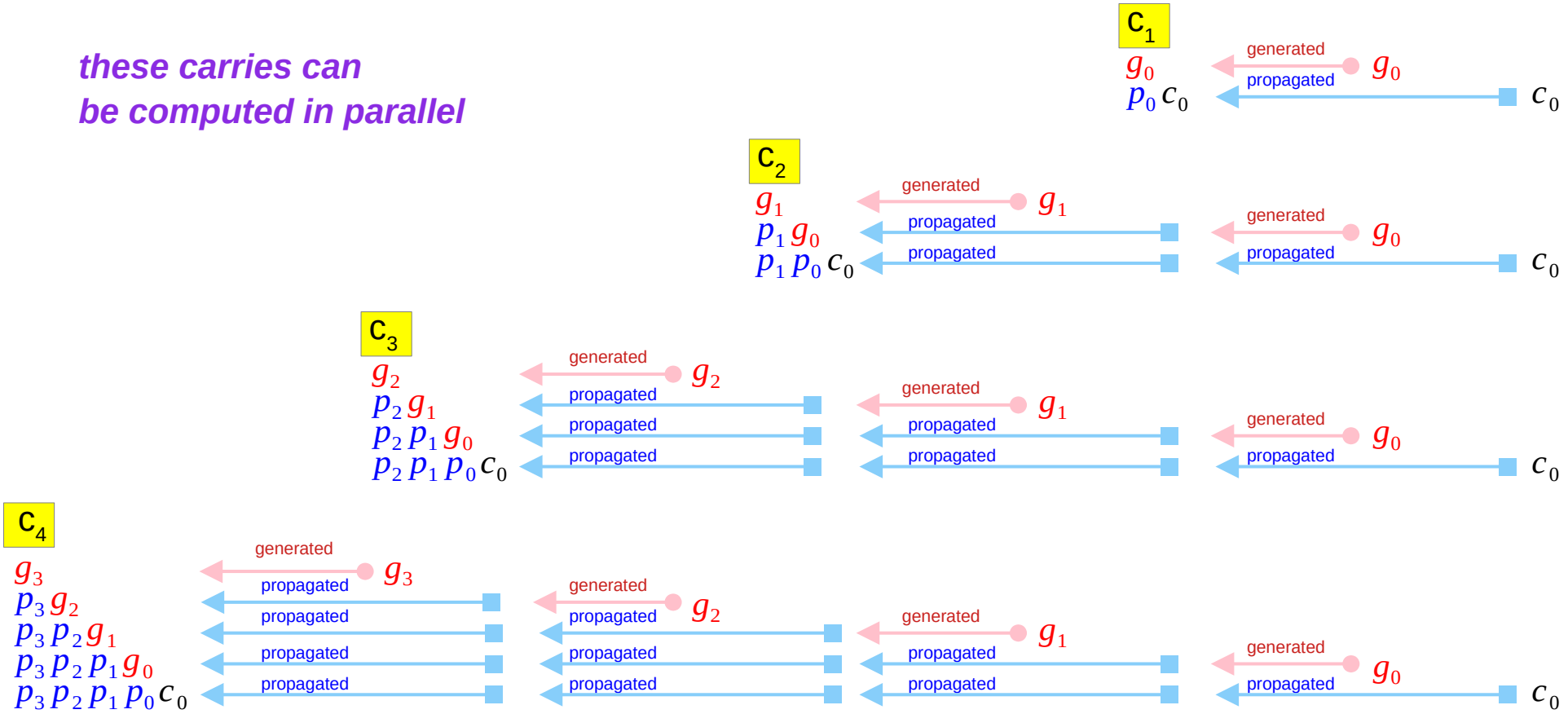
$$g_2 + p_2 g_1 + p_2 p_1 g_0 + p_2 p_1 p_0 c_0 = c_3$$

$$g_3 + p_3 g_2 + p_3 p_2 g_1 + p_3 p_2 p_1 g_0 + p_3 p_2 p_1 p_0 c_0 = c_4$$

Carry Generate and Carry Propagate Signals



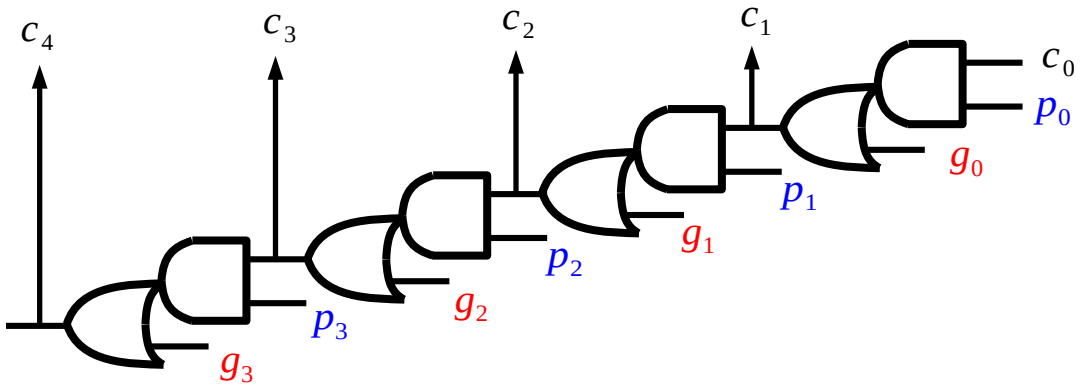
these carries can be computed in parallel



Carry equations into Gates (1)

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

$$c_4 = g_3 + p_3\left[g_2 + p_2\left[g_1 + p_1\left[g_0 + p_0 c_0\right]\right]\right]$$

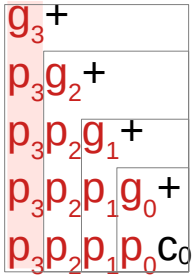


8 gate delay

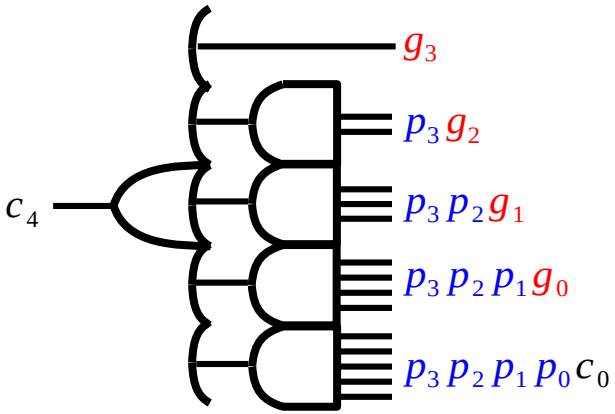
Fan-in number: small
Stage number: large

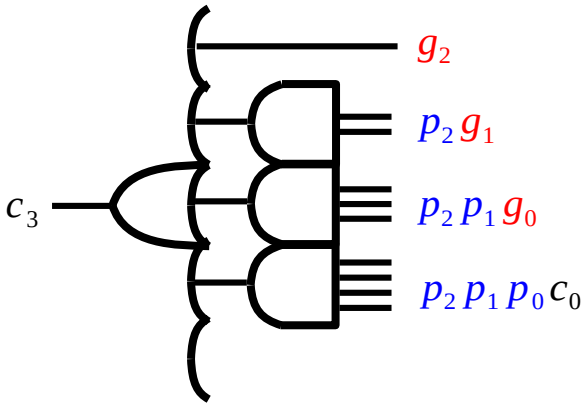
Ripple Carry Adder

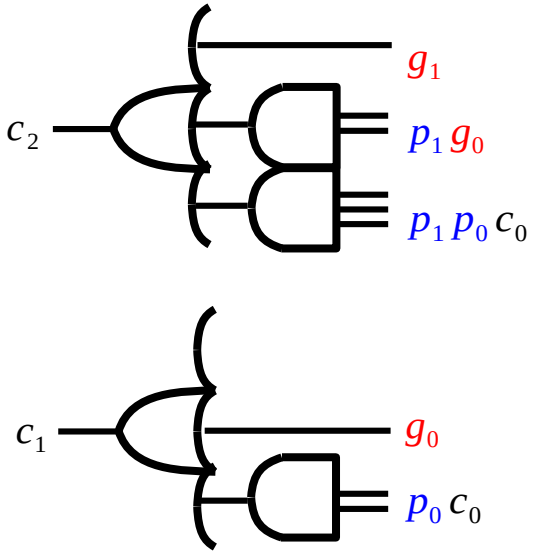
Carry equations into Gates (2)



$$g_0 + p_0 c_0 = c_1$$
$$g_1 + p_1 g_0 + p_1 p_0 c_0 = c_2$$
$$g_2 + p_2 g_1 + p_2 p_1 g_0 + p_2 p_1 p_0 c_0 = c_3$$
$$g_3 + p_3 g_2 + p_3 p_2 g_1 + p_3 p_2 p_1 g_0 + p_3 p_2 p_1 p_0 c_0 = c_4$$





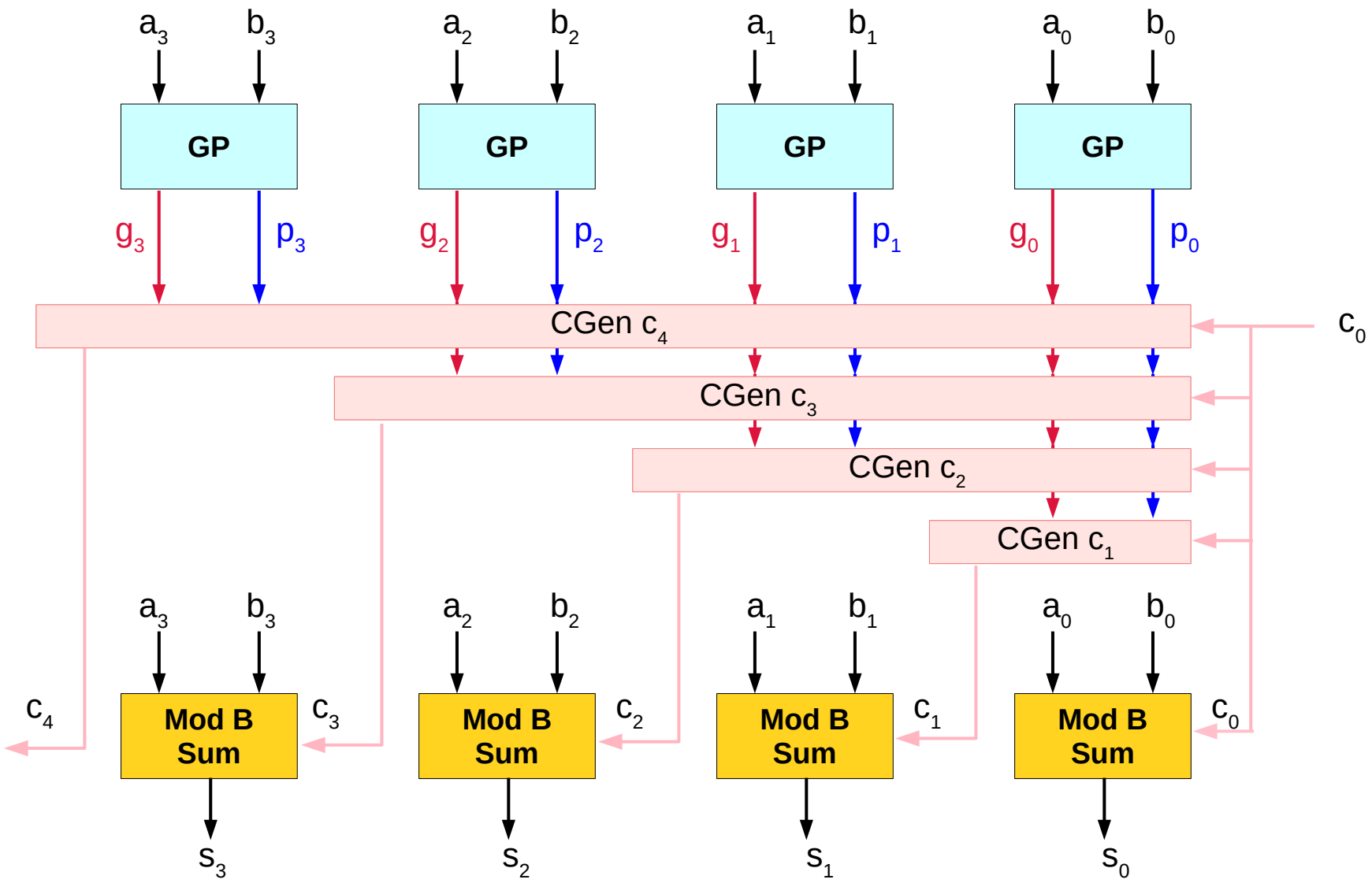


Sum of Product :
2 gate delay

Fan-in number: large
Stage number: small

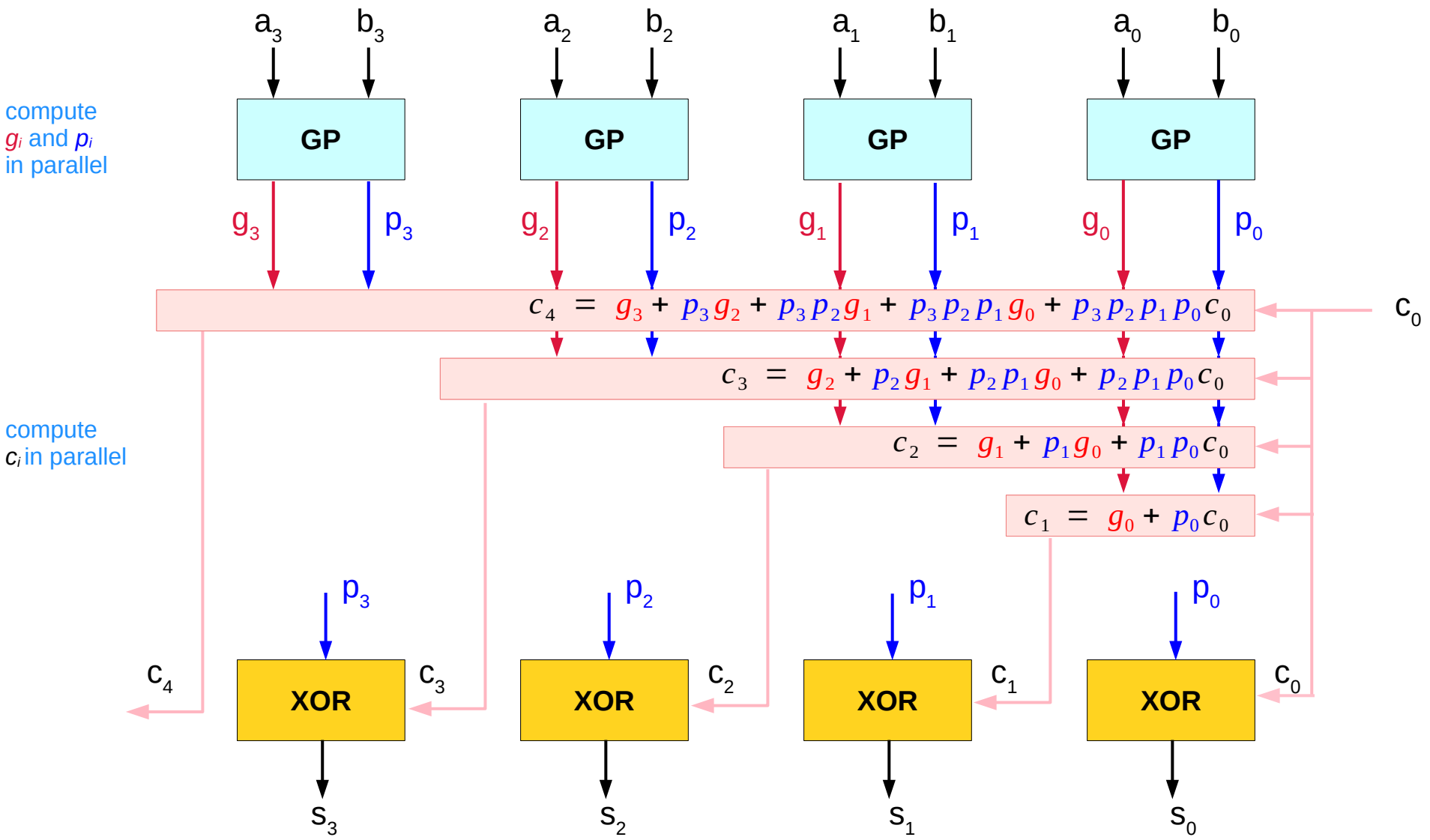
Carry Lookahead Adder

Carry Lookahead Adder



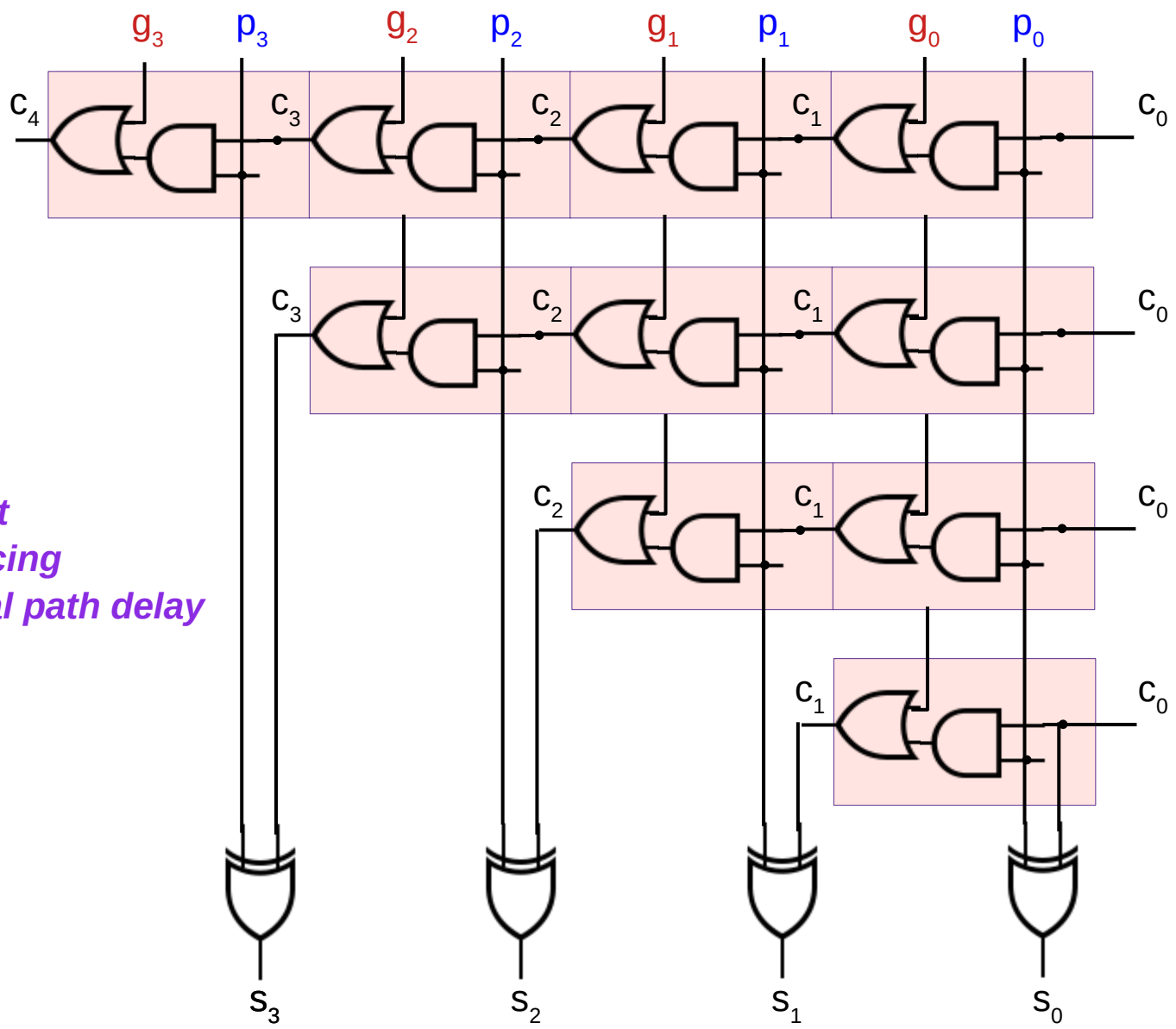
Synthesis of Arithmetic Circuits: FPGA, ASIC and Ebedded Systems, J-P Deschamps et al

Carry Lookahead Adder



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

Replicating carry circuits (1)



Half Adder

$p_i = a_i \oplus b_i$

$g_i = a_i \wedge b_i$

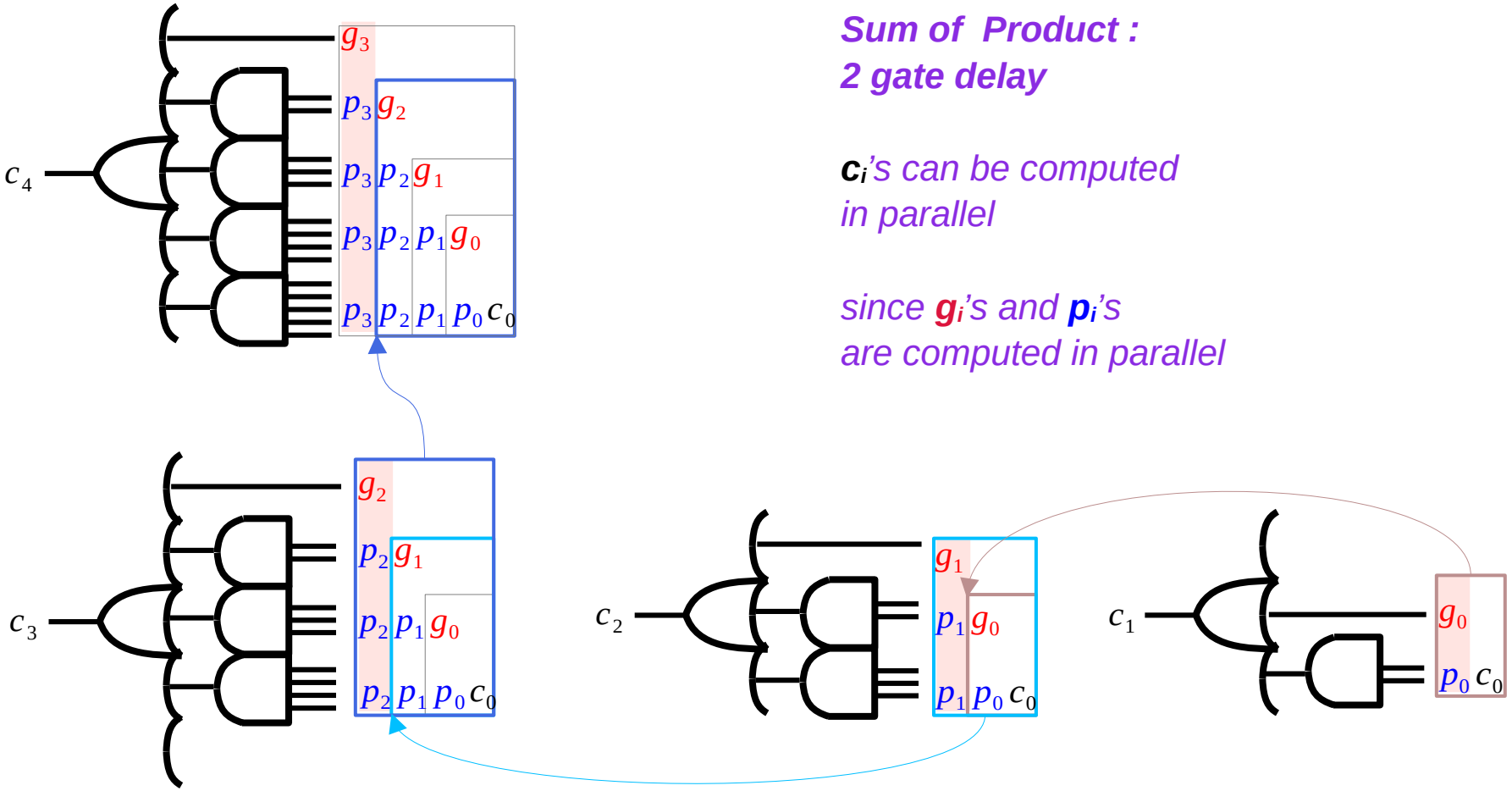
Half Adder

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

$s_i = p_i \oplus c_i$

no merit in reducing a critical path delay

Replicating carry circuits (2)



Sum of Product :
2 gate delay

c_i's can be computed
in parallel

since ***g_i***'s and ***p_i***'s
are computed in parallel

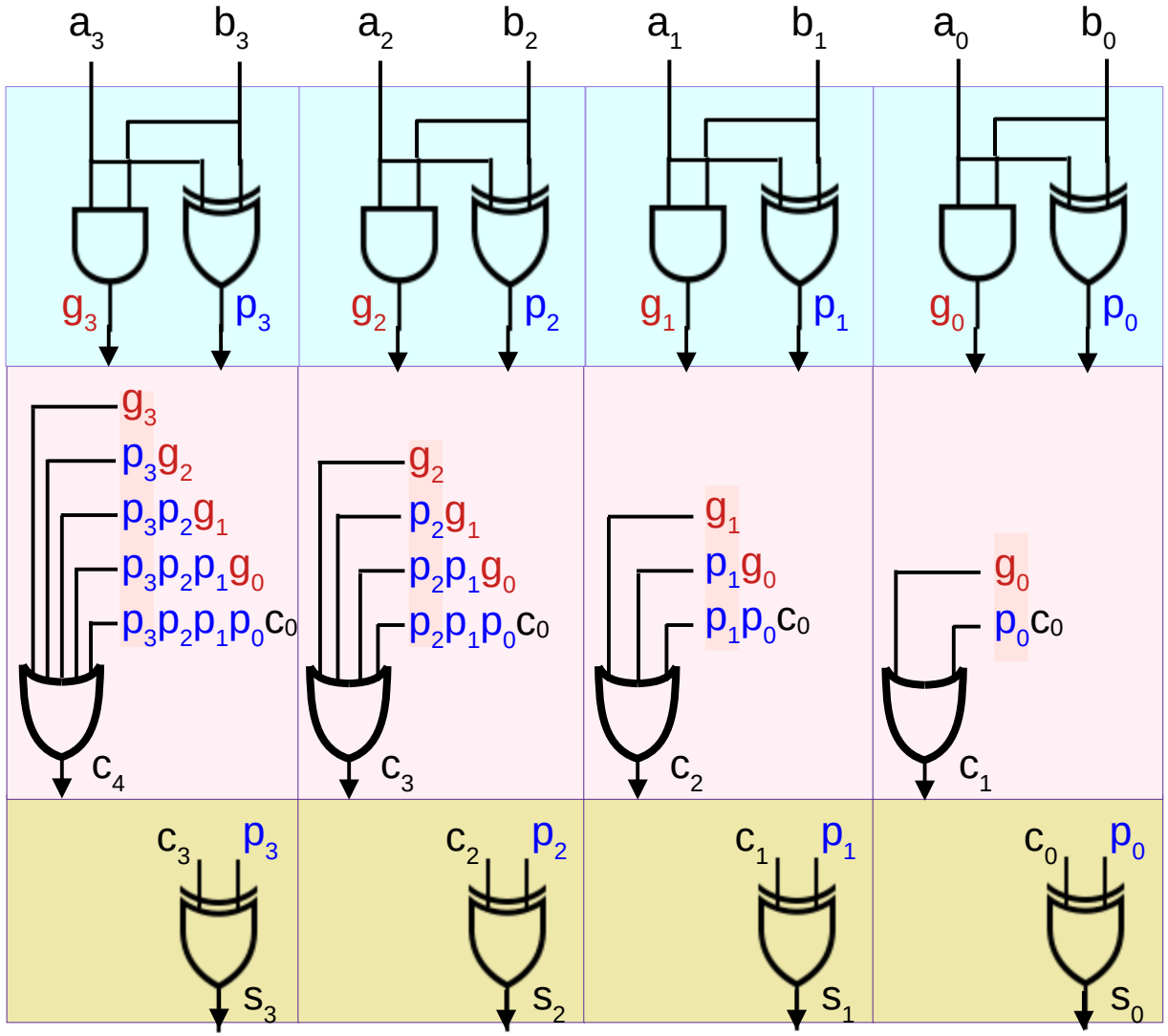
Limitations of Carry Lookahead

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	possible?	$= c_{31}$
AND33, OR33	possible?	$= c_{32}$

Large number of fan-in : *impractical*

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

A 4-bit Carry Lookahead Adder



compute g_i and p_i in parallel

compute c_i in parallel

compute S_i in parallel

GP

g_i : 1 gate delay ||
p_i : 1 gate delay

CGen

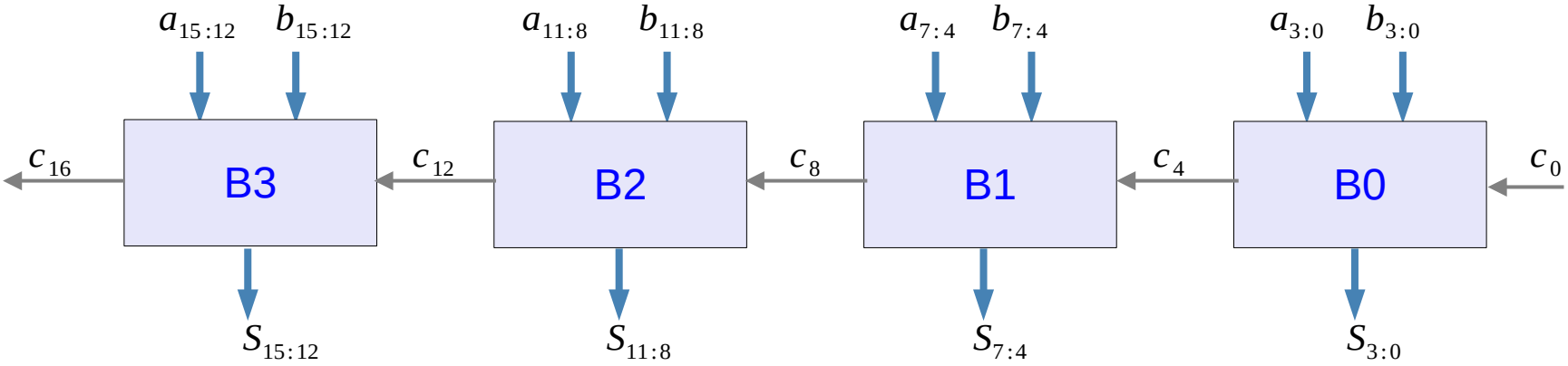
Sum of Product :
2 gate delay

c_i : 2 gate delay

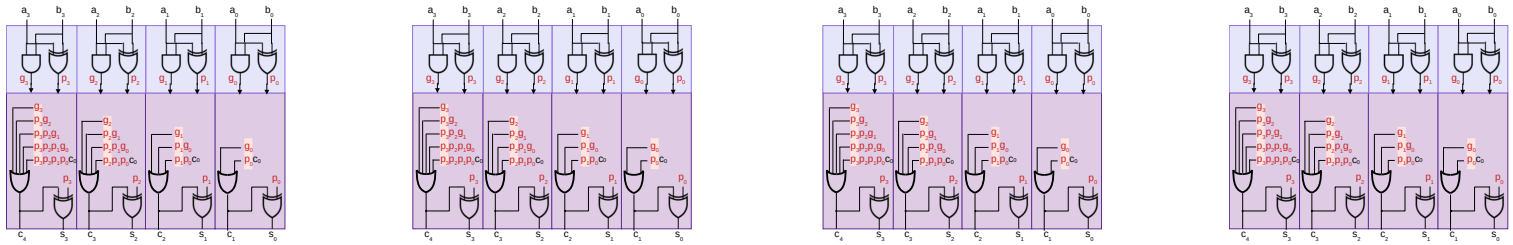
Sum

S_i : 1 gate delay

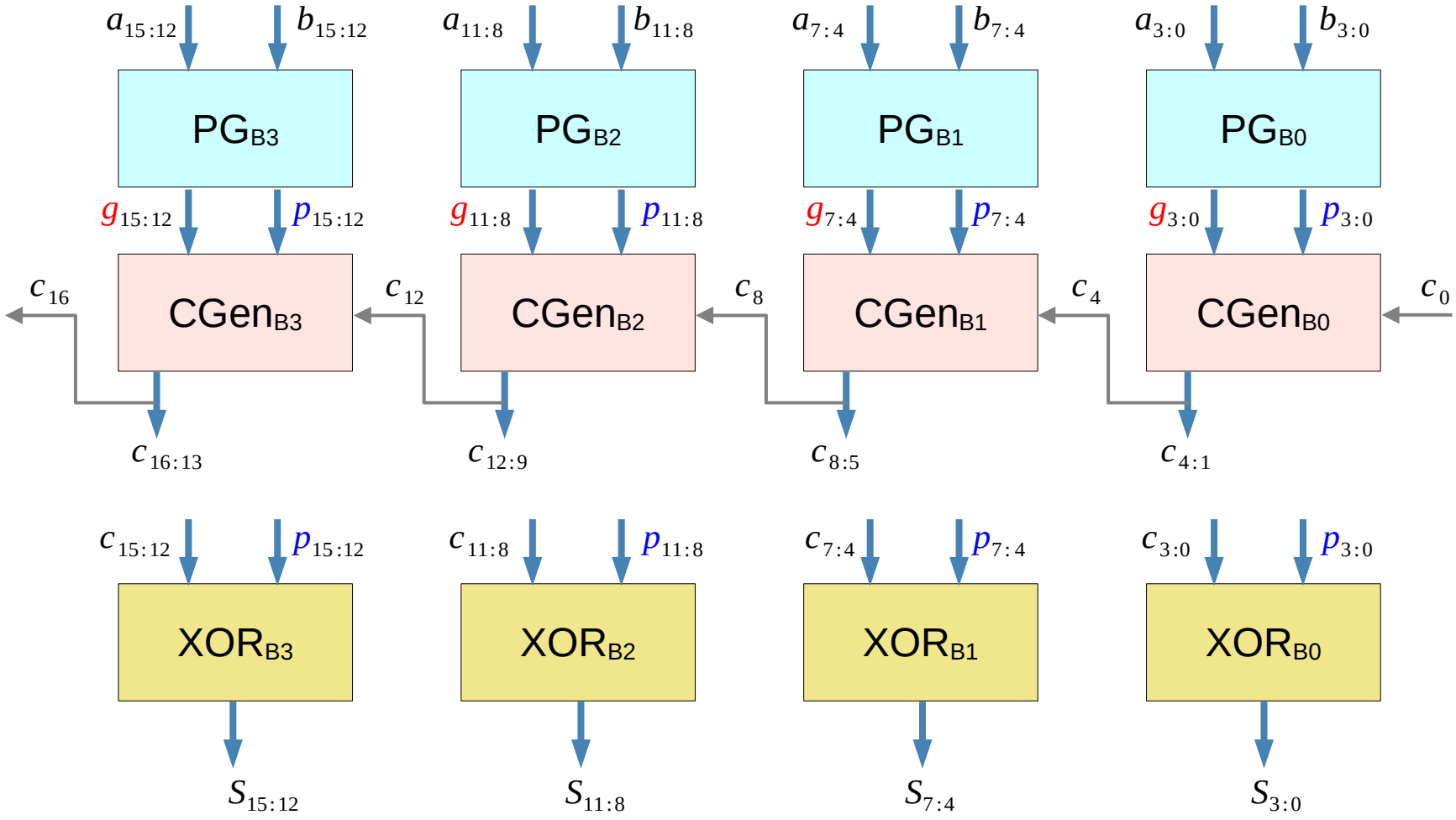
A 16-bit adder using four 4-bit CLA's



partition a 16-bit adder into four 4-bit **CLA** blocks : B3, B2, B1, B0



Block CLA Structures (1)



Block CLA Structures (2)

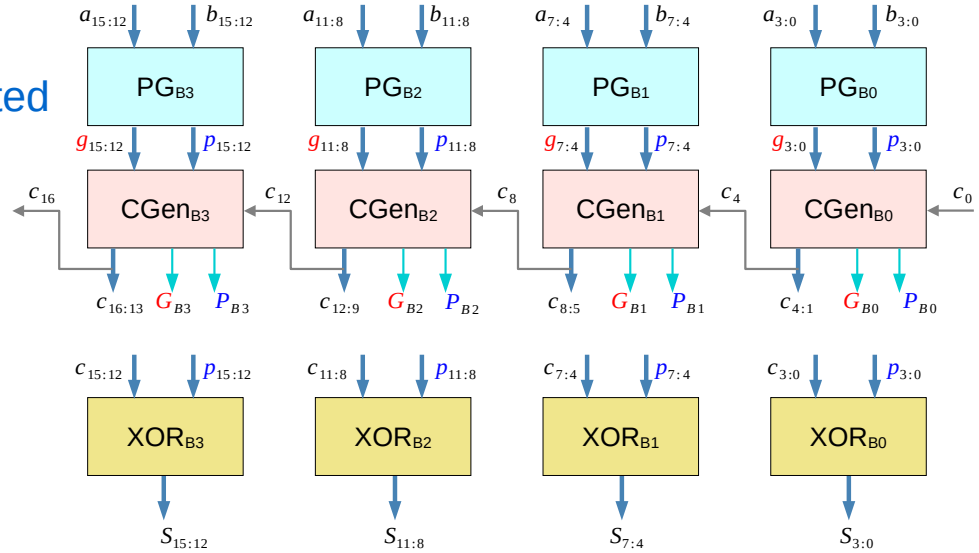
Each block computes 4 g's and 4 p's in parallel

As soon as c_{in}, 4 g's, and 4 p's of each block is available,
each block computes 4 carries in parallel

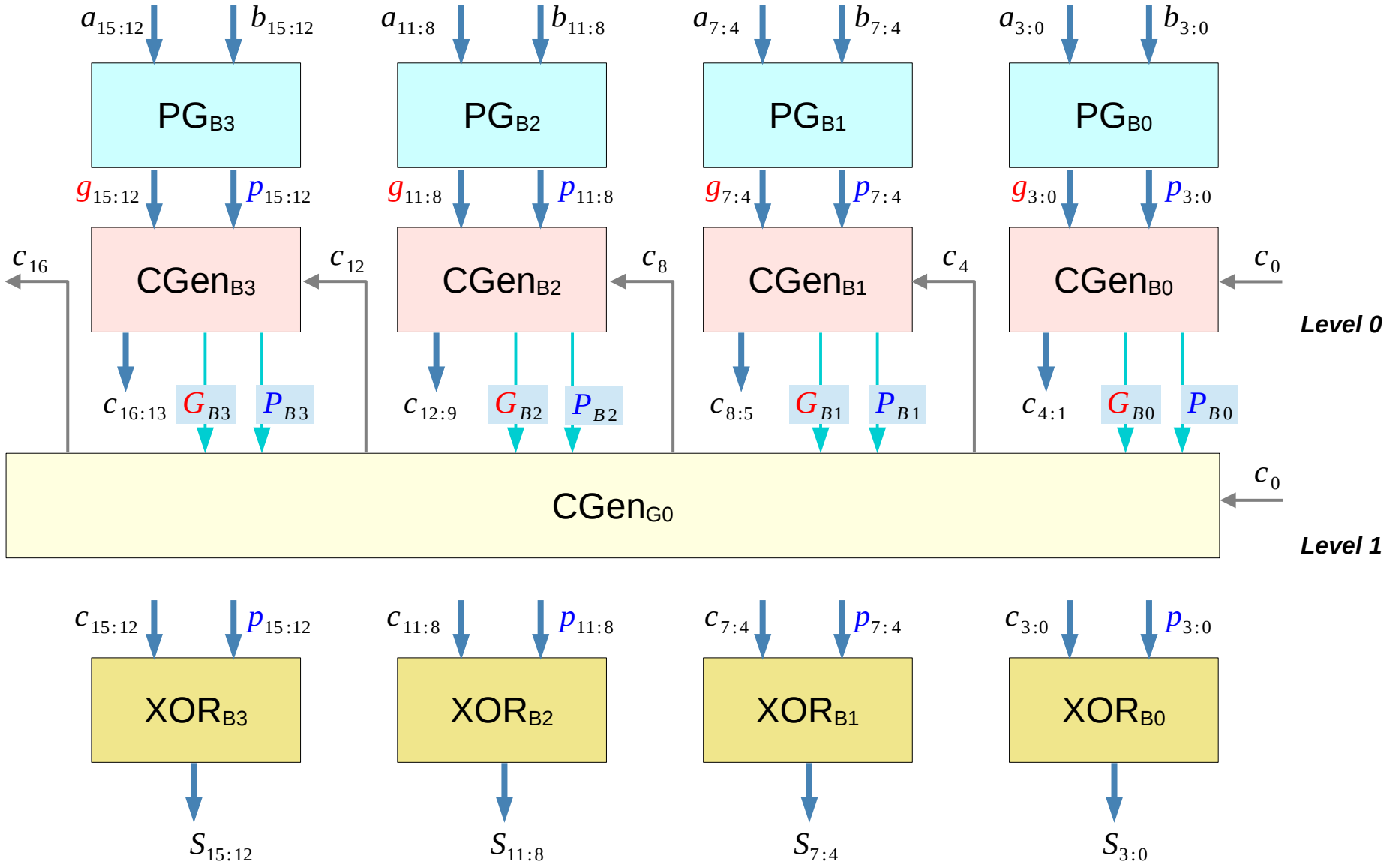
As soon as 4 carries and 4 p's of each block is available,
each block computes 4 sums in parallel

However,
only after c_{out} of B0/B1/B2 is available,
the four carries of B1/B2/B3 can be computed

These can be computed in parallel,
if block carry generate signal G_{Bi}
and block carry propagate signal P_{Bi}
are used

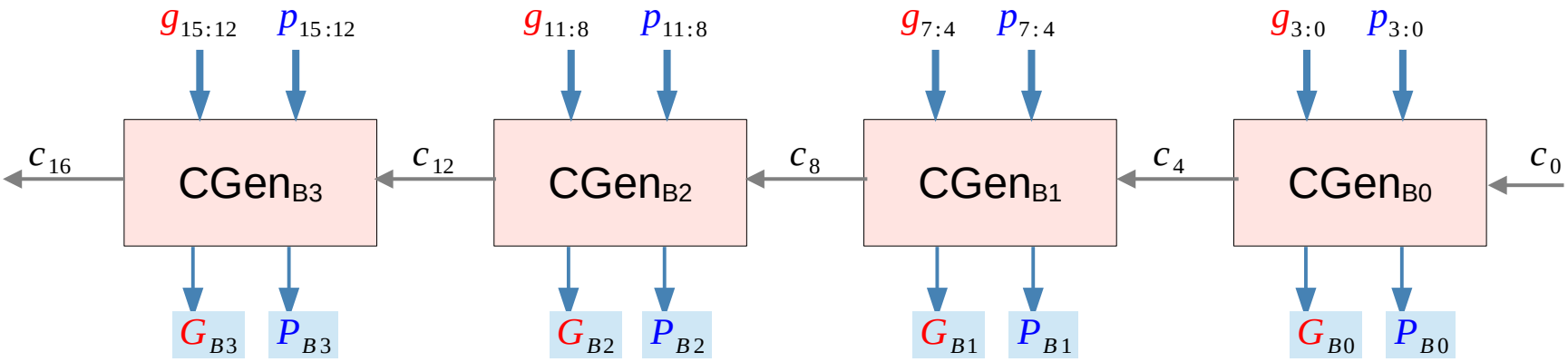


Multi-level CLA



Block Carry Equations

Let us focus on the carry circuits of each block



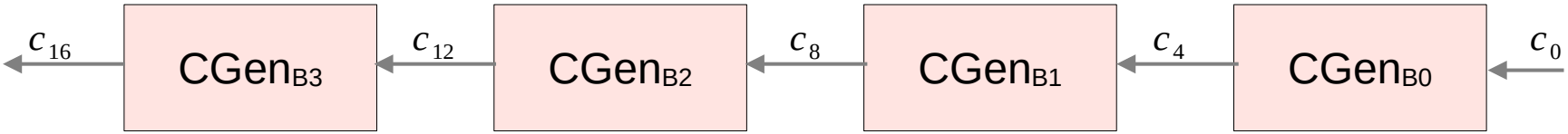
$c_{16} = G_{B3} + P_{B3} c_{12}$ $c_{12} = G_{B2} + P_{B2} c_8$ $c_8 = G_{B1} + P_{B1} c_4$ $c_4 = G_{B0} + P_{B0} c_0$

Define block G and block P signal to compute c_{16}, c_{12}, c_8, c_4 in parallel

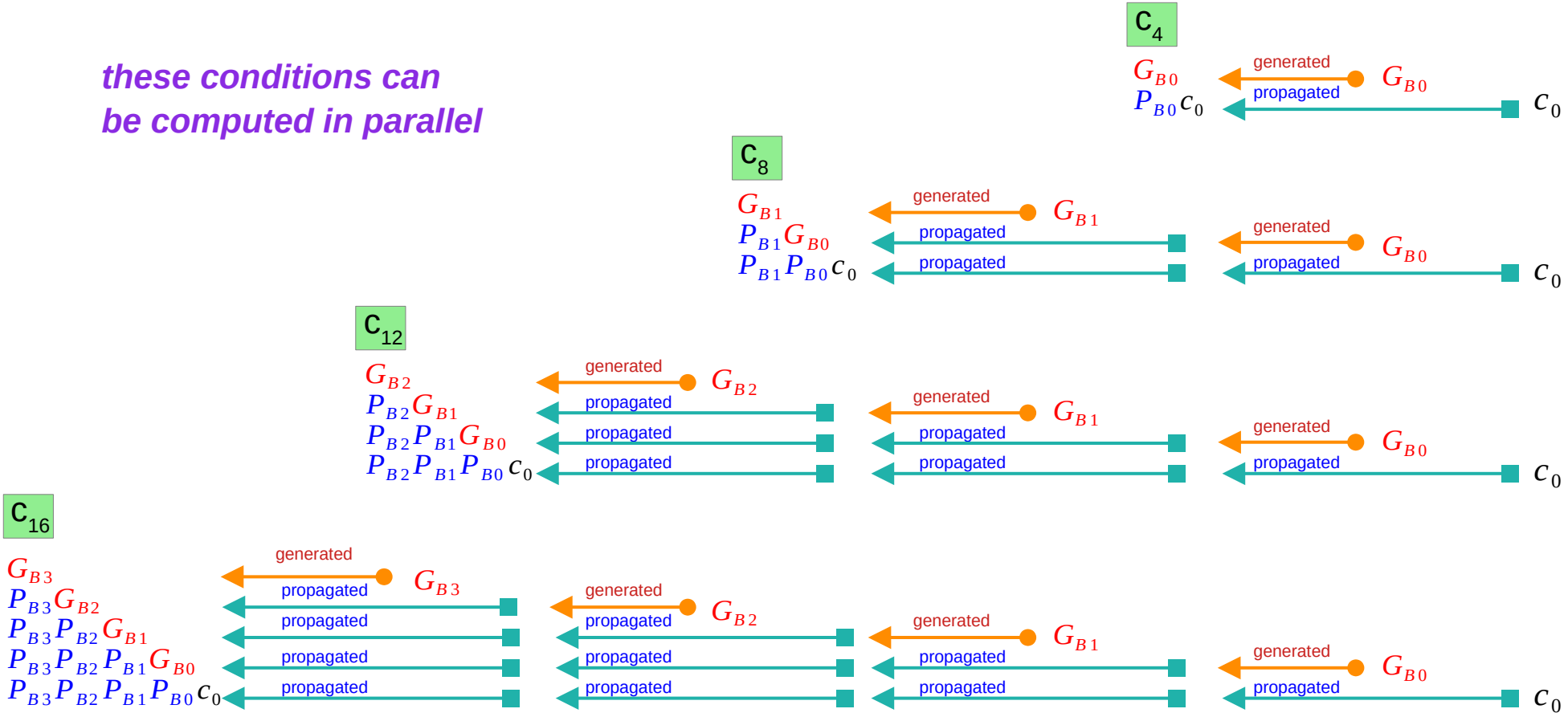
G_{Bi} *denotes the carries which are generated in the block* ***Bi***

$P_{Bi}c_{in}$ *denotes the input carry* c_{in} *which is propagated through the block* ***Bi***

Block Carry Generate and Propagate Signals



these conditions can be computed in parallel



Expanding Block Carry Equations

$$c_{4\cdot(i+1)} = G_{Bi} + P_{Bi} c_{4\cdot i}$$

$$c_4 = G_{B0} + P_{B0} c_0 \quad \Rightarrow \quad c_4 = G_{B0} + P_{B0} c_0$$

$$c_8 = G_{B1} + P_{B1} c_4 \quad \Rightarrow \quad c_8 = G_{B1} + P_{B1}\left[G_{B0} + P_{B0} c_0\right]$$

$$c_{12} = G_{B2} + P_{B2} c_8 \quad \Rightarrow \quad c_{12} = G_{B2} + P_{B2}\left[G_{B1} + P_{B1}\left[G_{B0} + P_{B0} c_0\right]\right]$$

$$c_{16} = G_{B3} + P_{B3} c_{12} \quad \Rightarrow \quad c_{16} = G_{B3} + P_{B3}\left[G_{B2} + P_{B2}\left[G_{B1} + P_{B1}\left[G_{B0} + P_{B0} c_0\right]\right]\right]$$

c_{out} c_{in}

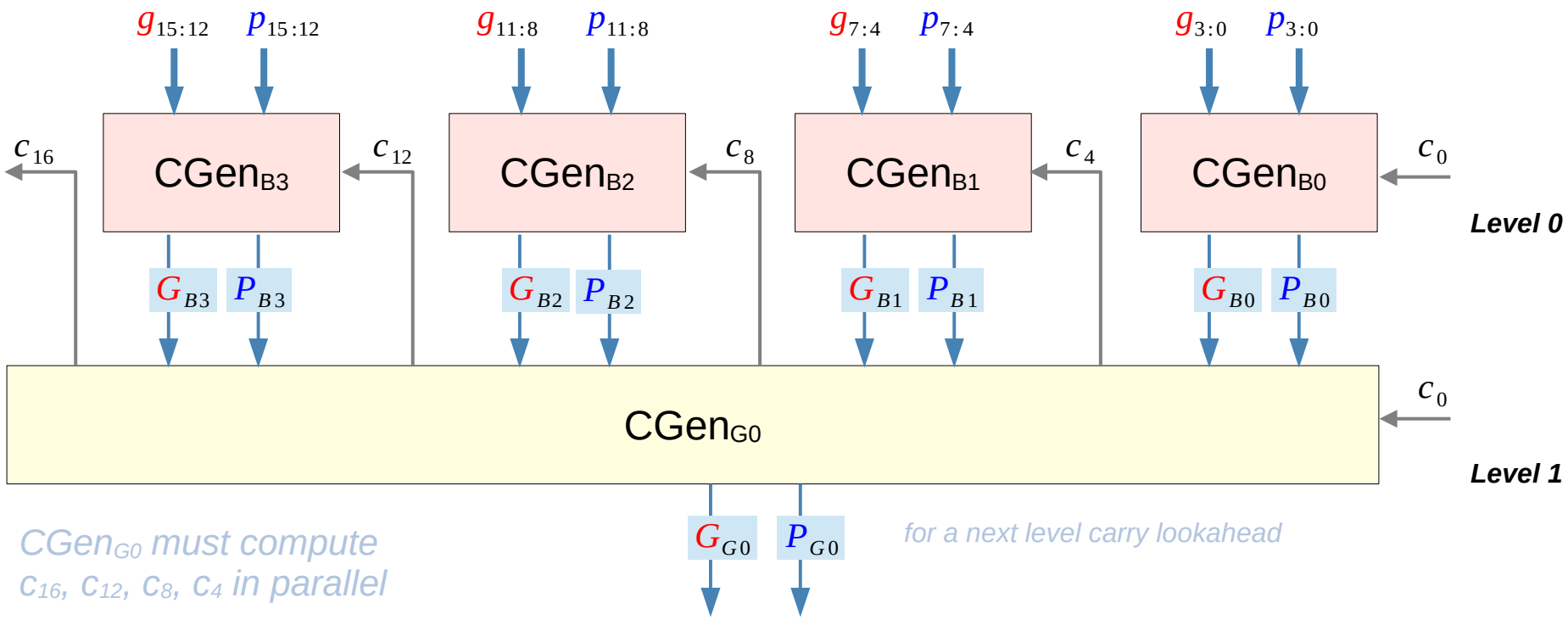
$$G_{B0} + P_{B0} c_0 = c_4$$

$$G_{B1} + P_{B1} G_{B0} + P_{B1} P_{B0} c_0 = c_8$$

$$G_{B2} + P_{B2} G_{B1} + P_{B2} P_{B1} G_{B0} + P_{B2} P_{B1} P_{B0} c_0 = c_{12}$$

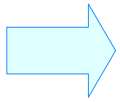
$$G_{B3} + P_{B3} G_{B2} + P_{B3} P_{B2} G_{B1} + P_{B3} P_{B2} P_{B1} G_{B0} + P_{B3} P_{B2} P_{B1} P_{B0} c_0 = c_{16}$$

Block Carry Generating Circuits



Carry Lookahead Logic Comparison

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



Block B0 Carries

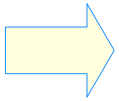
$$g_0 + p_0 c_0 = c_1$$

$$g_1 + p_1 g_0 + p_1 p_0 c_0 = c_2$$

$$g_2 + p_2 g_1 + p_2 p_1 g_0 + p_2 p_1 p_0 c_0 = c_3$$

$$g_3 + p_3 g_2 + p_3 p_2 g_1 + p_3 p_2 p_1 g_0 + p_3 p_2 p_1 p_0 c_0 = c_4$$

$$c_{4\cdot(i+1)} = G_{Bi} + P_{Bi} c_{4\cdot i}$$



Group G0 Carries

$$G_{B0} + P_{B0} c_0 = c_4$$

$$G_{B1} + P_{B1} G_{B0} + P_{B1} P_{B0} c_0 = c_8$$

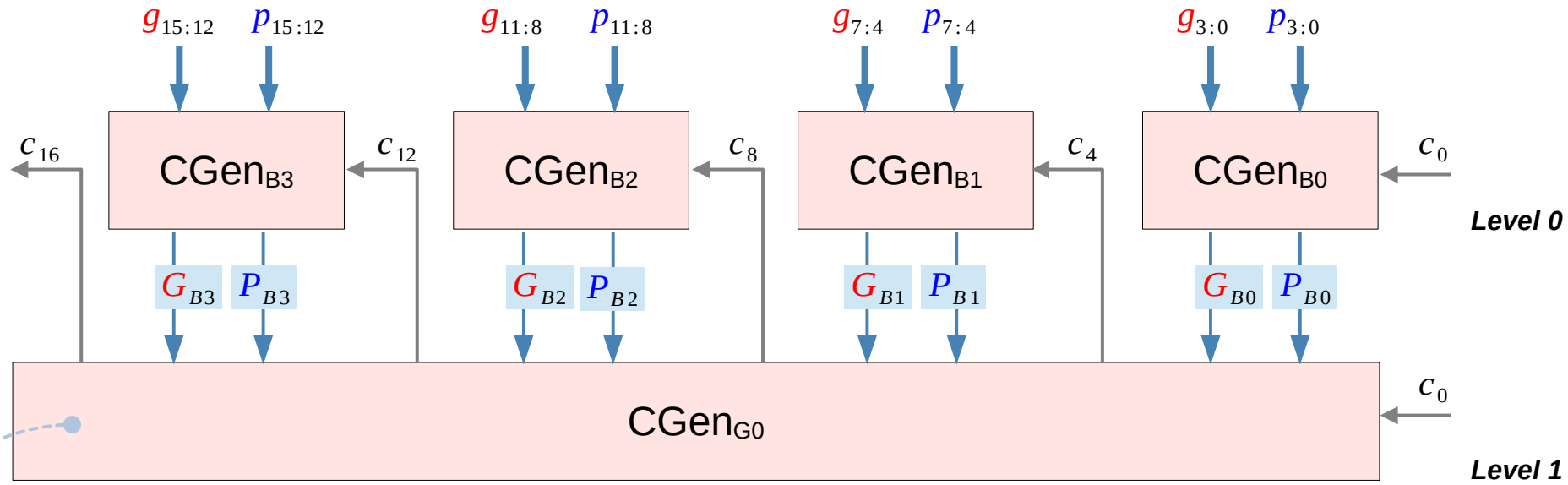
$$G_{B2} + P_{B2} G_{B1} + P_{B2} P_{B1} G_{B0} + P_{B2} P_{B1} P_{B0} c_0 = c_{12}$$

$$G_{B3} + P_{B3} G_{B2} + P_{B3} P_{B2} G_{B1} + P_{B3} P_{B2} P_{B1} G_{B0} + P_{B3} P_{B2} P_{B1} P_{B0} c_0 = c_{16}$$

the same structure

Block Carry Generating Circuits – simplified

Let us focus on the carry circuits of each block



has the same structure as the structure of each block

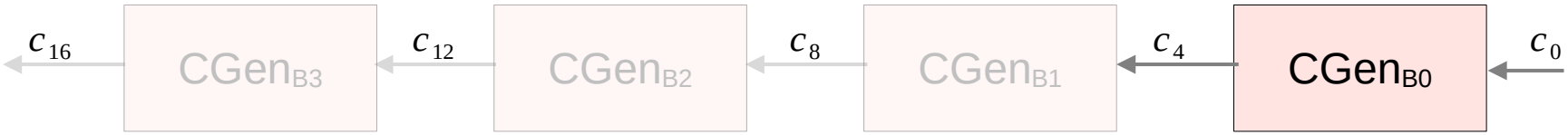
$$G_{B0} + P_{B0}c_0 = c_4$$

$$G_{B1} + P_{B1}G_{B0} + P_{B1}P_{B0}c_0 = c_8$$

$$G_{B2} + P_{B2}G_{B1} + P_{B2}P_{B1}G_{B0} + P_{B2}P_{B1}P_{B0}c_0 = c_{12}$$

$$G_{B3} + P_{B3}G_{B2} + P_{B3}P_{B2}G_{B1} + P_{B3}P_{B2}P_{B1}G_{B0} + P_{B3}P_{B2}P_{B1}P_{B0}c_0 = c_{16}$$

G_{B0}, P_{B0} : Block Carry Generate and Propagate

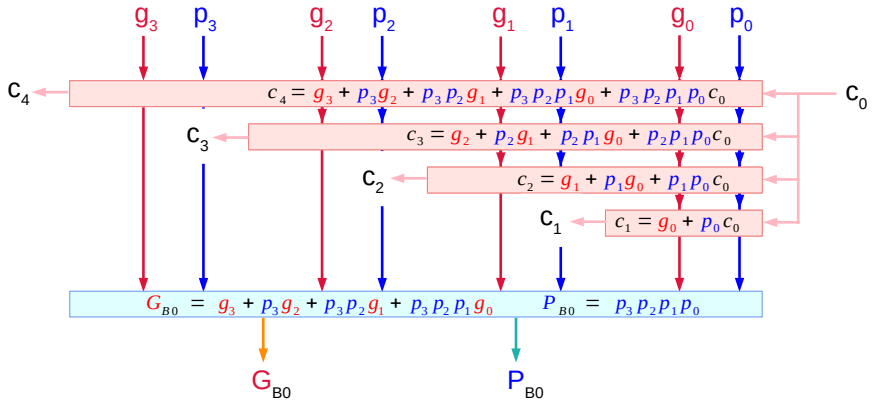


$$c_4 = g_3 + p_3 g_2 + p_3 p_2 g_1 + p_3 p_2 p_1 g_0 + p_3 p_2 p_1 p_0 c_0$$

$$G_{B0} = g_3 + p_3 g_2 + p_3 p_2 g_1 + p_3 p_2 p_1 g_0$$

$$P_{B0} = p_3 p_2 p_1 p_0$$

$$c_4 = G_{B0} + P_{B0} c_0$$

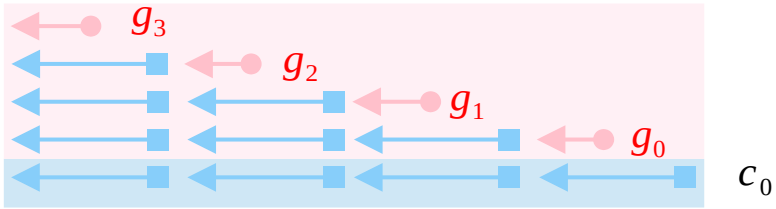


G_{B0} *denotes the generated carries in the block B0*

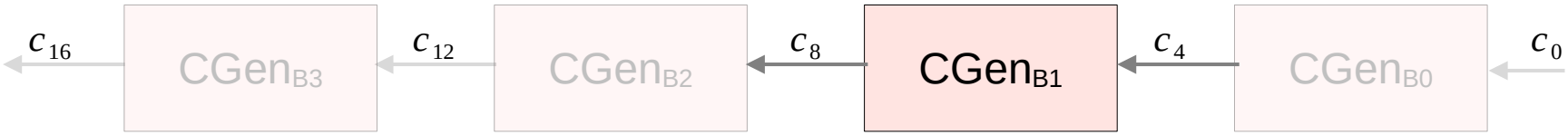
$P_{B0} c_0$ *denotes the the propagated input carry* c_0

c_4

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



G_{B1}, P_{B1} : Block Carry Generate and Propagate

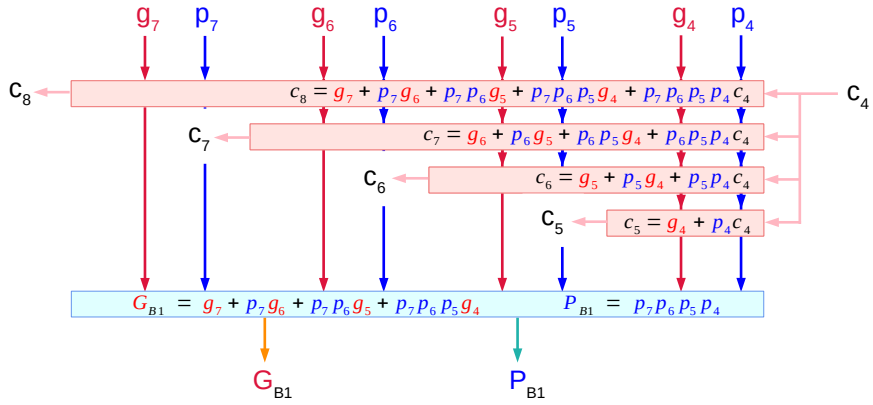


$$c_8 = g_7 + p_7 g_6 + p_7 p_6 g_5 + p_7 p_6 p_5 g_4 + p_7 p_6 p_5 p_4 c_4$$

$$G_{B1} = g_7 + p_7 g_6 + p_7 p_6 g_5 + p_7 p_6 p_5 g_4$$

$$P_{B1} = p_7 p_6 p_5 p_4$$

$$c_8 = G_{B1} + P_{B1} c_4$$

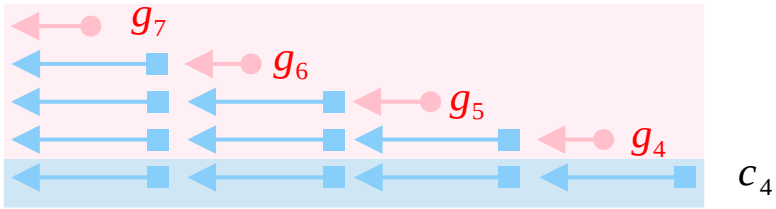


G_{B1} *denotes the generated carries in the block B1*

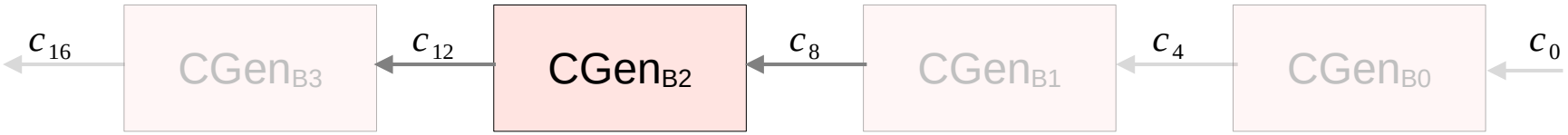
$P_{B1} c_4$ *denotes the the propagated input carry* c_4

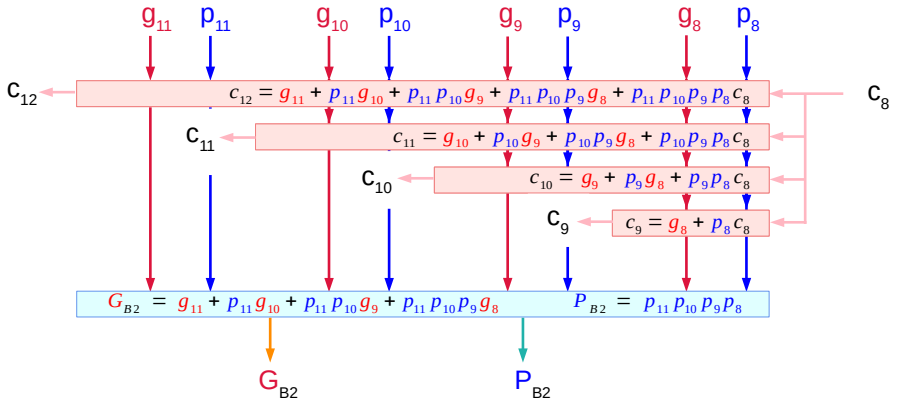
c_8

g_7
$p_7 g_6$
$p_7 p_6 g_5$
$p_7 p_6 p_5 g_4$
$p_7 p_6 p_5 p_4 c_4$



G_{B2}, P_{B2} : Block Carry Generate and Propagate



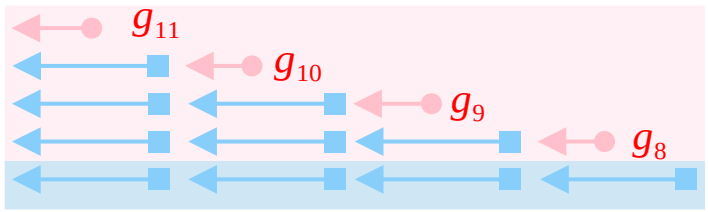


$$c_{12} = g_{11} + p_{11}g_{10} + p_{11}p_{10}g_9 + p_{11}p_{10}p_9g_8 + p_{11}p_{10}p_9p_8c_8$$

$$G_{B2} = g_{11} + p_{11}g_{10} + p_{11}p_{10}g_9 + p_{11}p_{10}p_9g_8$$

$$P_{B2} = p_{11}p_{10}p_9p_8$$

$$c_{12} = G_{B2} + P_{B2}c_8$$



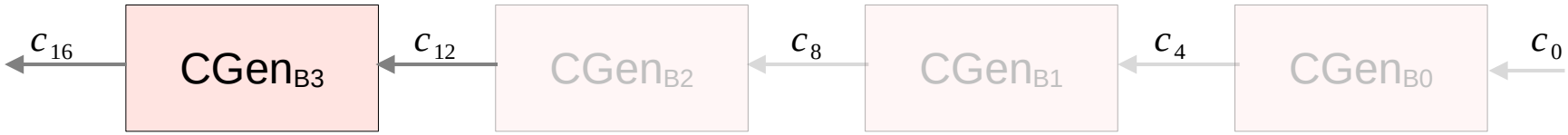
c_{12}

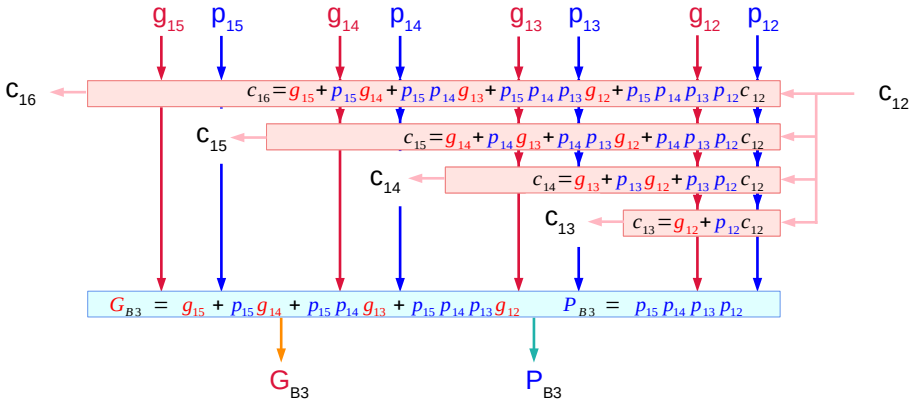
g_{11}
$p_{11}g_{10}$
$p_{11}p_{10}g_9$
$p_{11}p_{10}p_9g_8$
$p_{11}p_{10}p_9p_8c_8$

G_{B2} *denotes the generated carries in the block B2*

$P_{B2}c_8$ *denotes the the propagated input carry* c_8

G_{B3}, P_{B3} : Block Carry Generate and Propagate



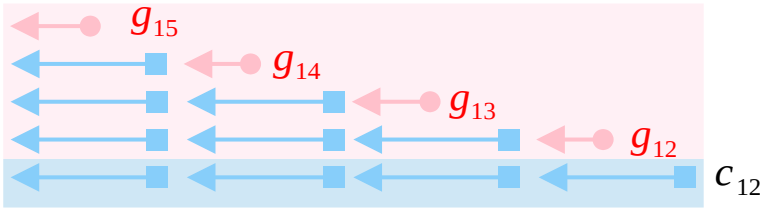


$$c_{16} = g_{15} + p_{15} g_{14} + p_{15} p_{14} g_{13} + p_{15} p_{14} p_{13} g_{12} + p_{15} p_{14} p_{13} p_{12} c_{12}$$

$$G_{B3} = g_{15} + p_{15} g_{14} + p_{15} p_{14} g_{13} + p_{15} p_{14} p_{13} g_{12}$$

$$P_{B3} = p_{15} p_{14} p_{13} p_{12}$$

$$c_{16} = G_{B3} + P_{B3} c_{12}$$



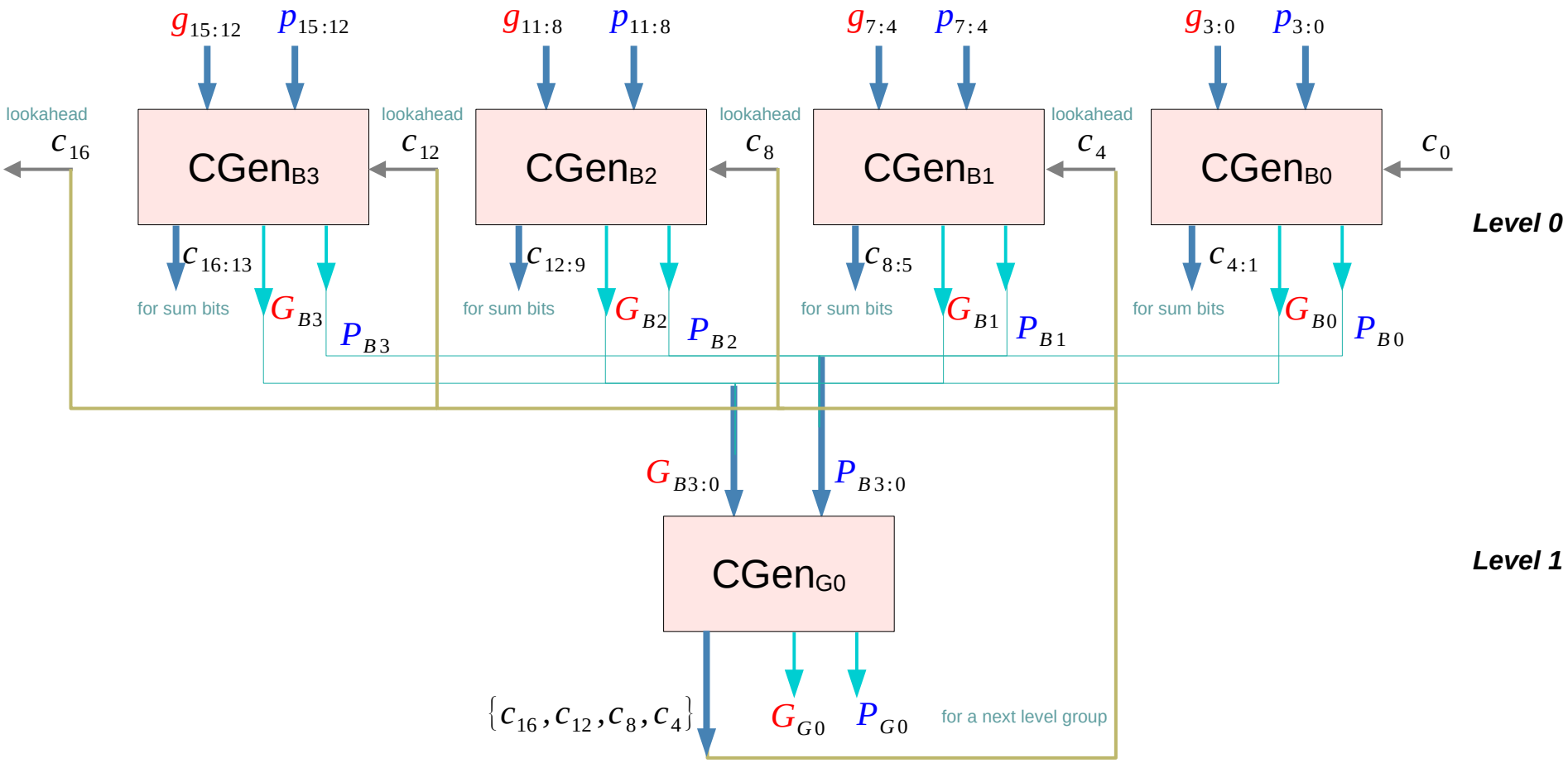
c_{16}

g_{15}
$p_{15} g_{14}$
$p_{15} p_{14} g_{13}$
$p_{15} p_{14} p_{13} g_{12}$
$p_{15} p_{14} p_{13} p_{12} c_{12}$

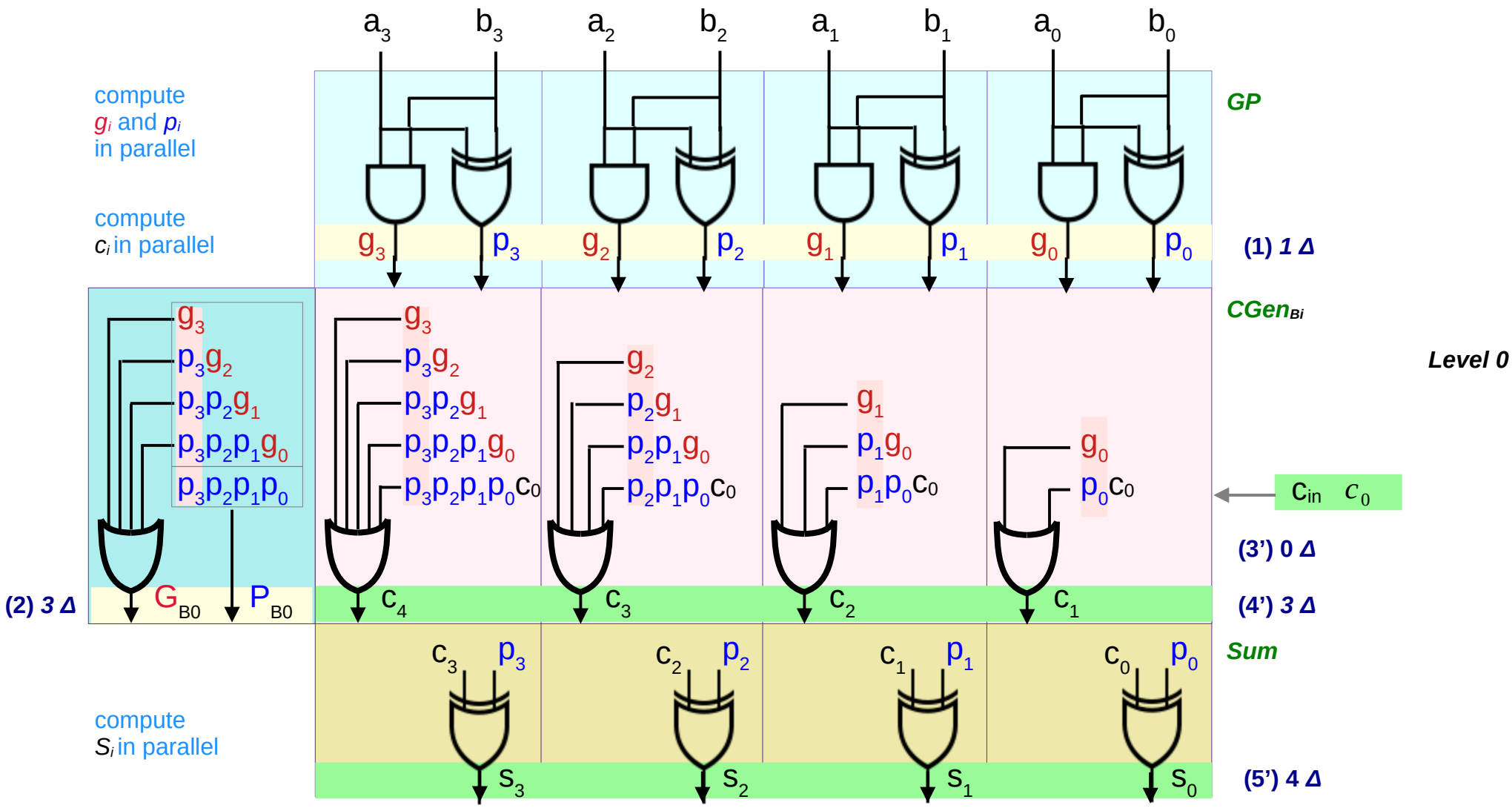
G_{B3} *denotes the generated carries in the block B3*

$P_{B2} c_8$ *denotes the the propagated input carry* c_8

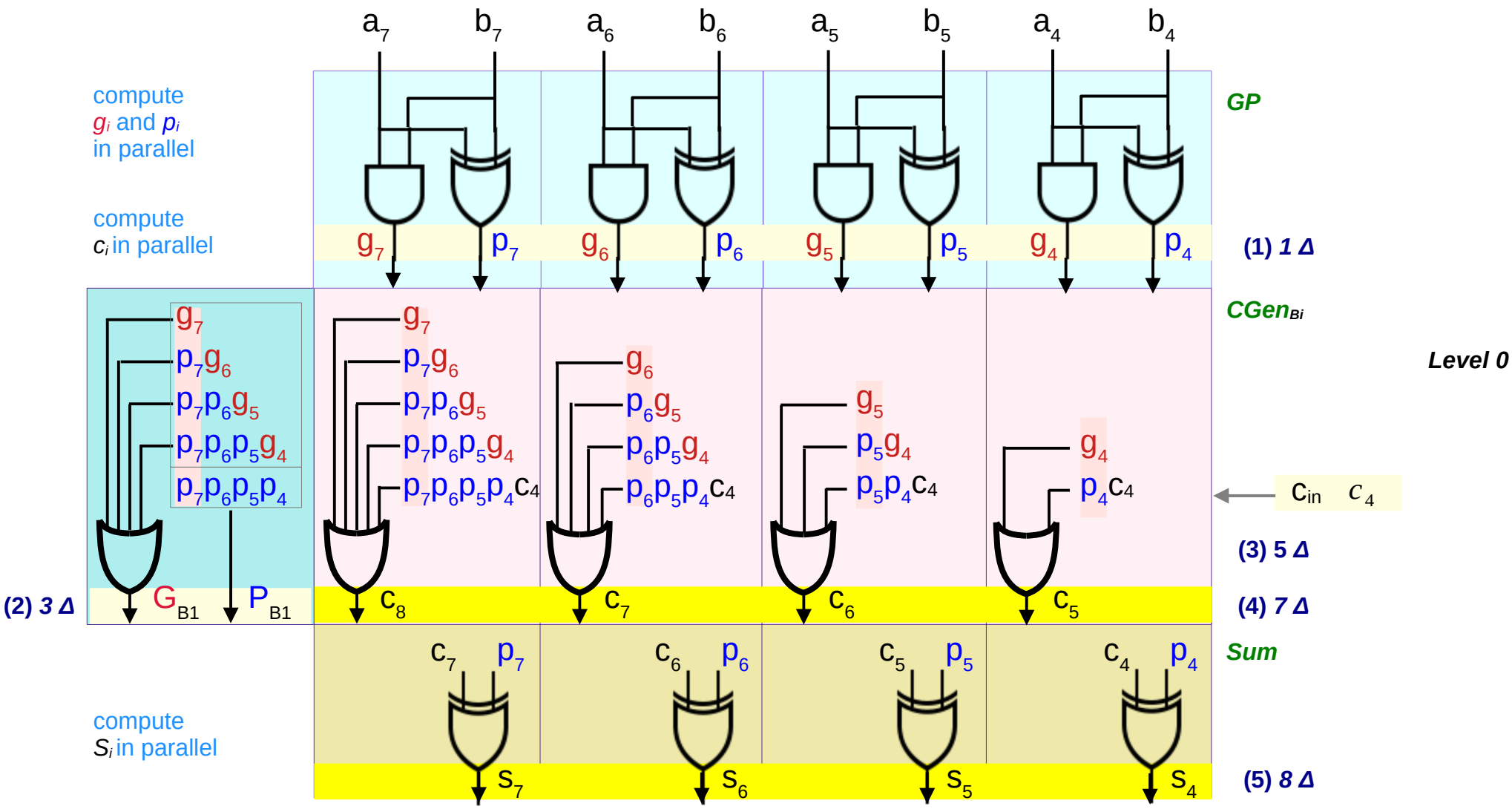
Block Carry Generating Circuits – detailed



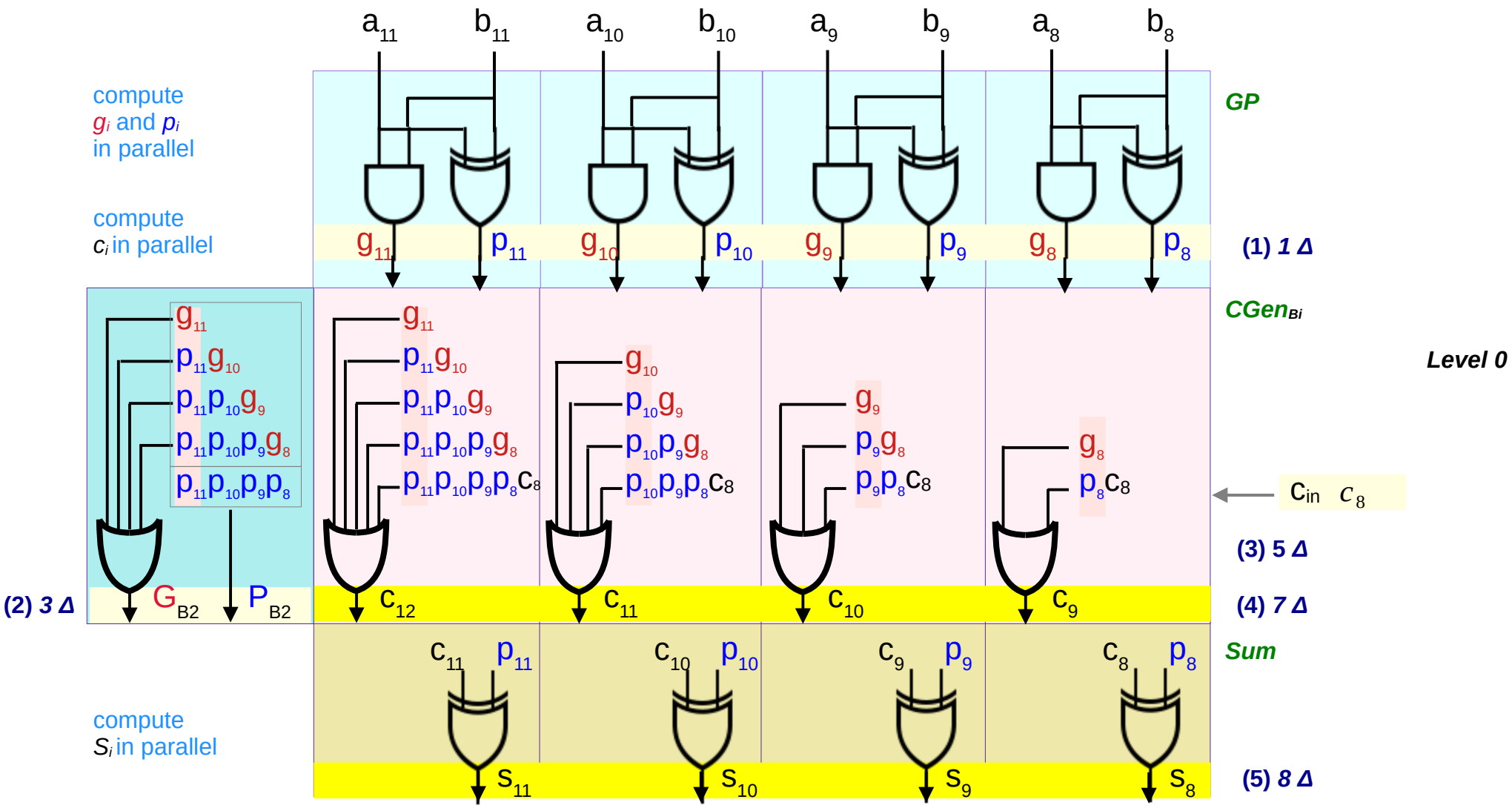
Simple Carry Lookahead Adder – B0



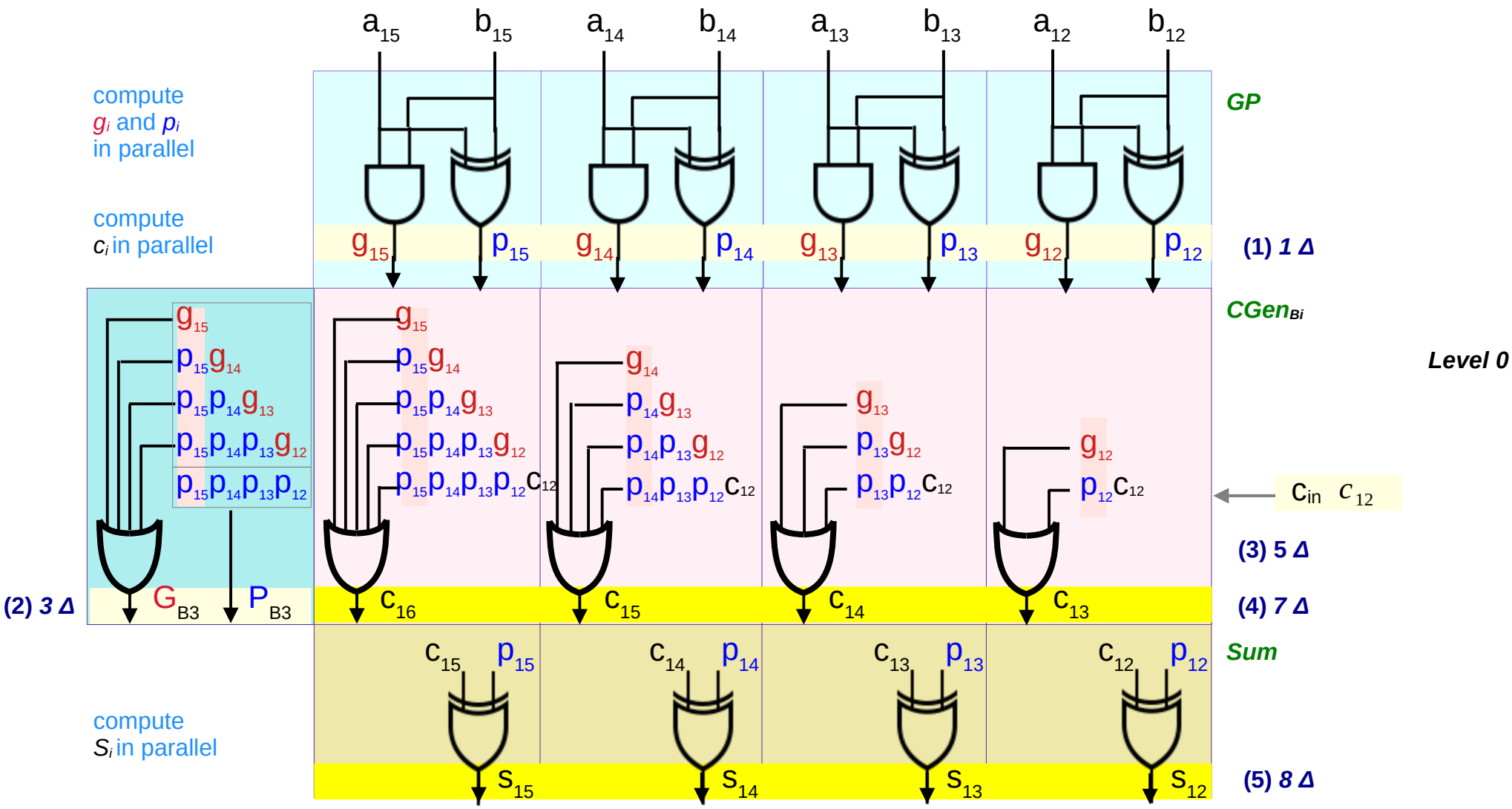
Simple Carry Lookahead Adder – B1



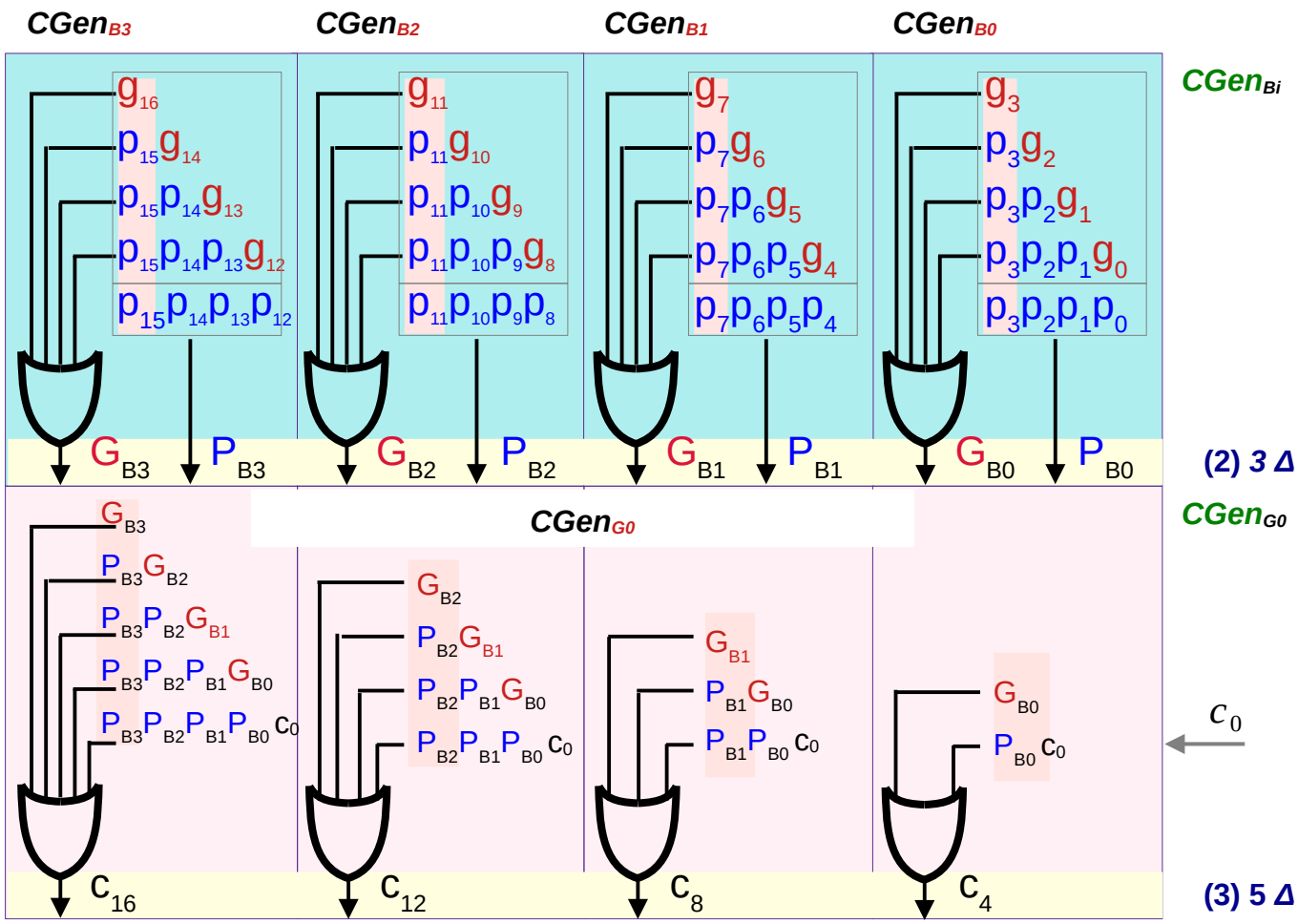
Simple Carry Lookahead Adder – B2



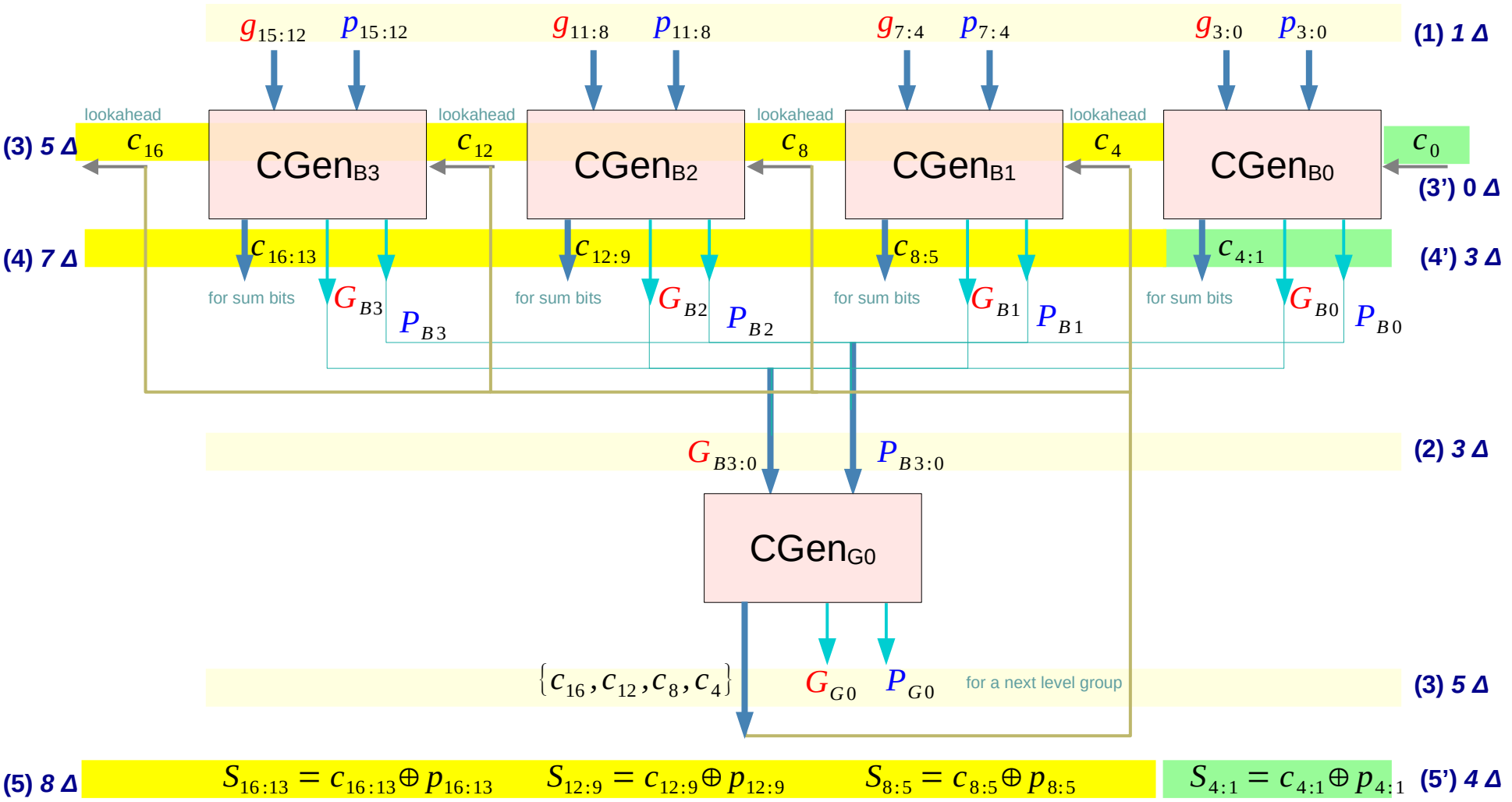
Simple Carry Lookahead Adder – B3



A 2-Level Carry Lookahead Adder – Level 1



Delay calculations of 2-Level CLA (1)



Delay calculations of 2-Level CLA (2)

Starting at time of zero:

- calculation of p_i and g_i is done at time 1,
- calculation of the P_{Bi} is done at time 2,
- calculation of the G_{Bi} is done at time 3,
- calculation of the inputs for the $CGen_{Bi}$ from the $CGen_{G0}$ are done at:
  - time 0 for $CGen_{B0}$,
  - time 5 for $CGen_{B1}$, $CGen_{B2}$, $CGen_{B3}$,
- calculation of the S_i are done at:
  - time 4 for Sum_{B0},
  - time 8 for Sum_{B1}, Sum_{B2}, Sum_{B3},
- calculation of the final carry bit (c_{16}) is done at time 5.

The maximal time is 8 gate delays (for $S_{[4:15]}$).

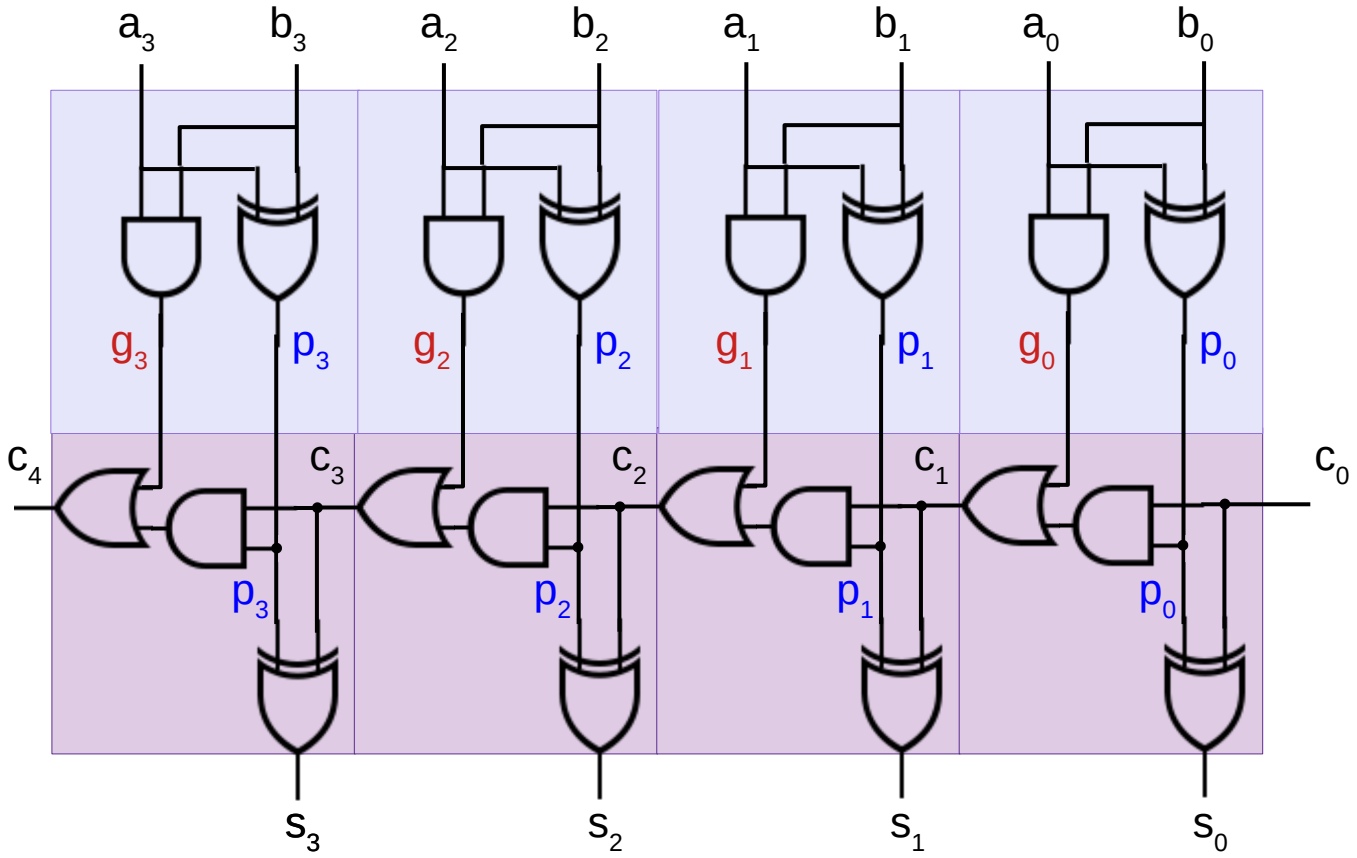
https://en.wikipedia.org/wiki/Carry-lookahead_adder

A Variation of 2 Level CLA

Ripple Carry Adder
+ Block Carry Lookahead

S. & D. Harris, Digital Design and Computer Architecture 2nd ed.

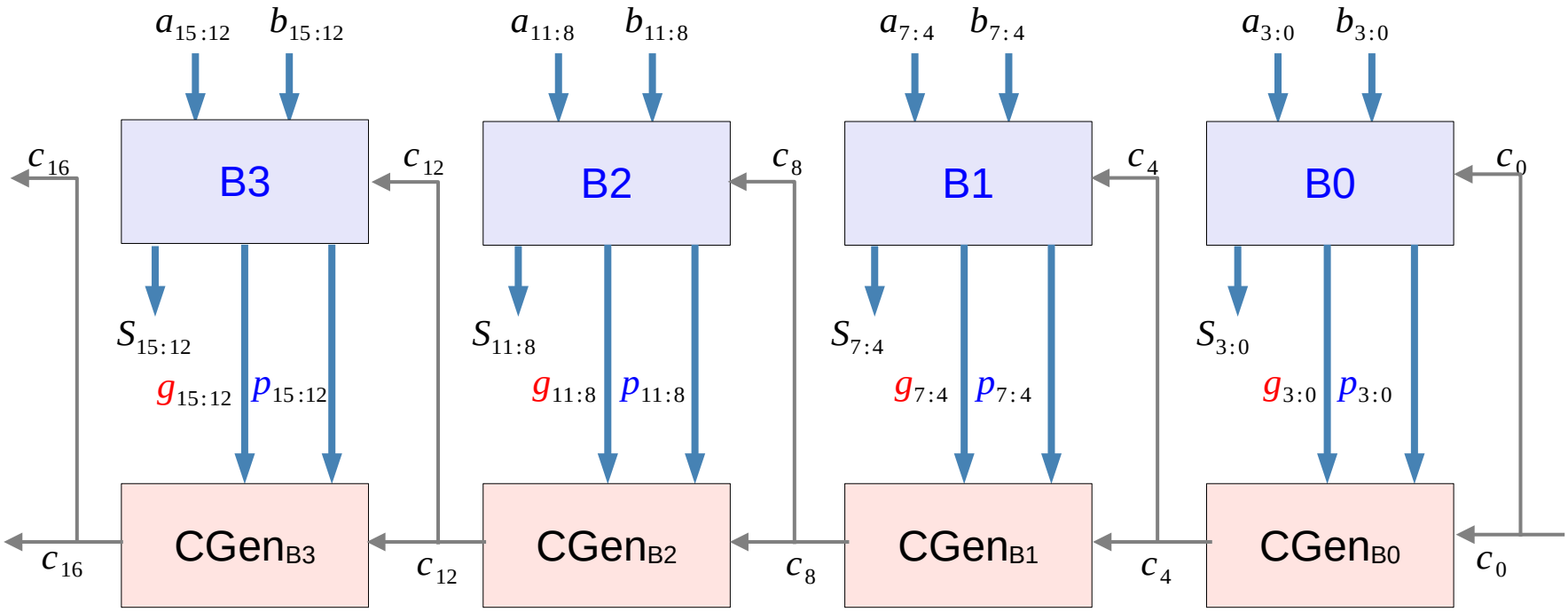
A 4-bit Ripple Carry Adder



Critical Path : $c_4 - c_3 - c_2 - c_1 - c_0$: 8 gate delay

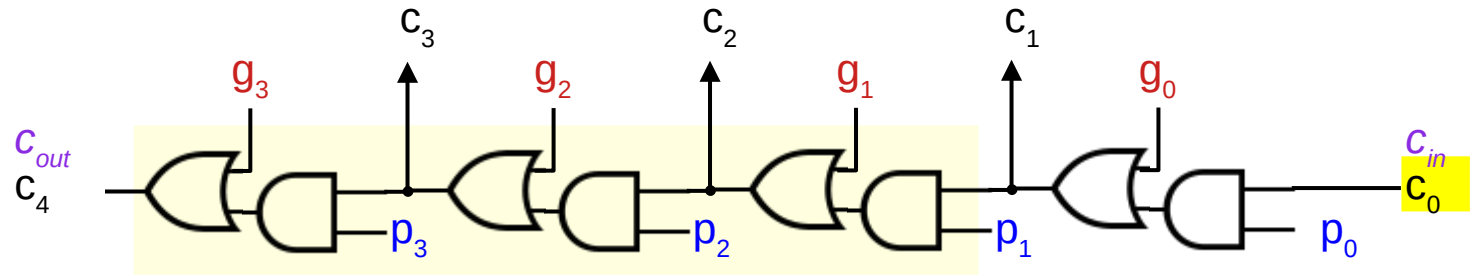
https://upload.wikimedia.org/wikiversity/en/1/18/RCA.Note.H.1.20151215.pdf

A 16-bit adder using four 4-bit CLA's

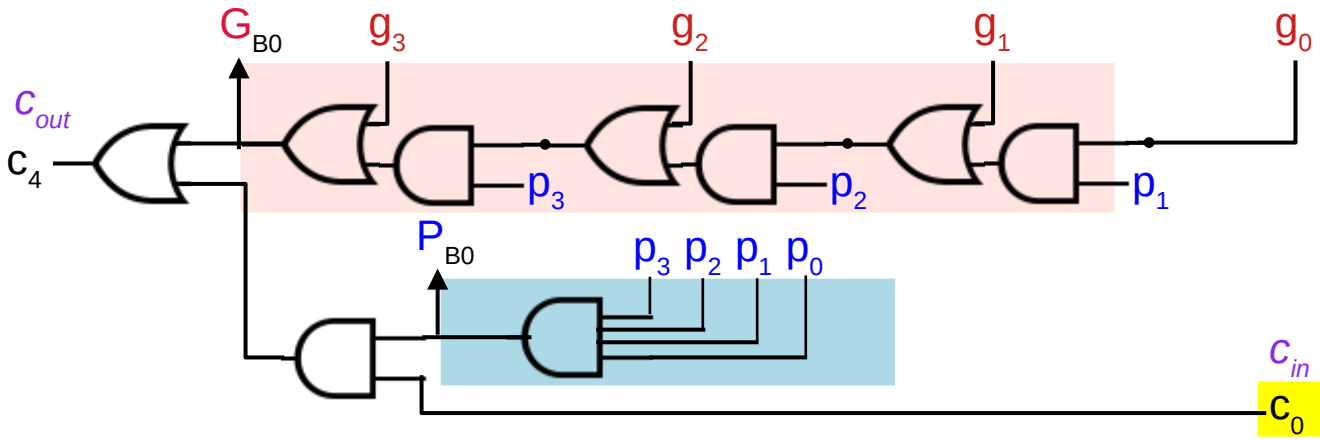


A 4-bit Ripple Carry Adder

$$c_4 = g_3 + p_3 g_2 + p_3 p_2 g_1 + p_3 p_2 p_1 g_0 + p_3 p_2 p_1 p_0 c_0$$



carry paths in the 4 FA's



carry lookahead circuits

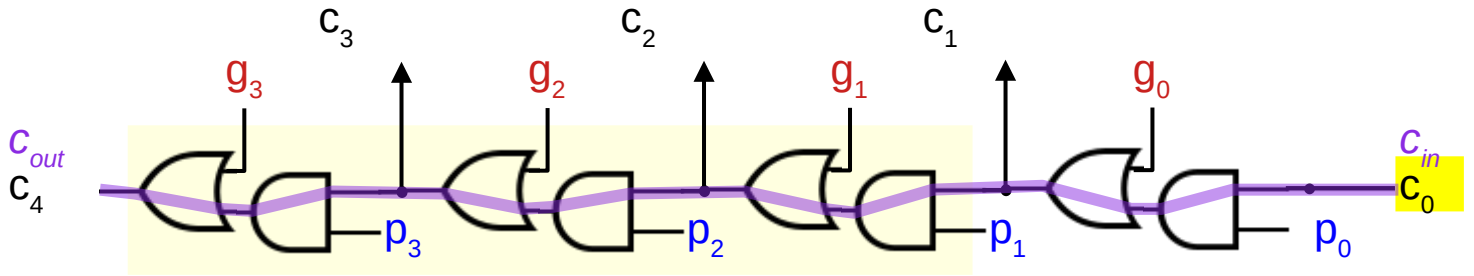
$$c_4 = G_{B0} + P_{B0} c_0$$

$$G_{B0} = g_3 + p_3 g_2 + p_3 p_2 g_1 + p_3 p_2 p_1 g_0$$

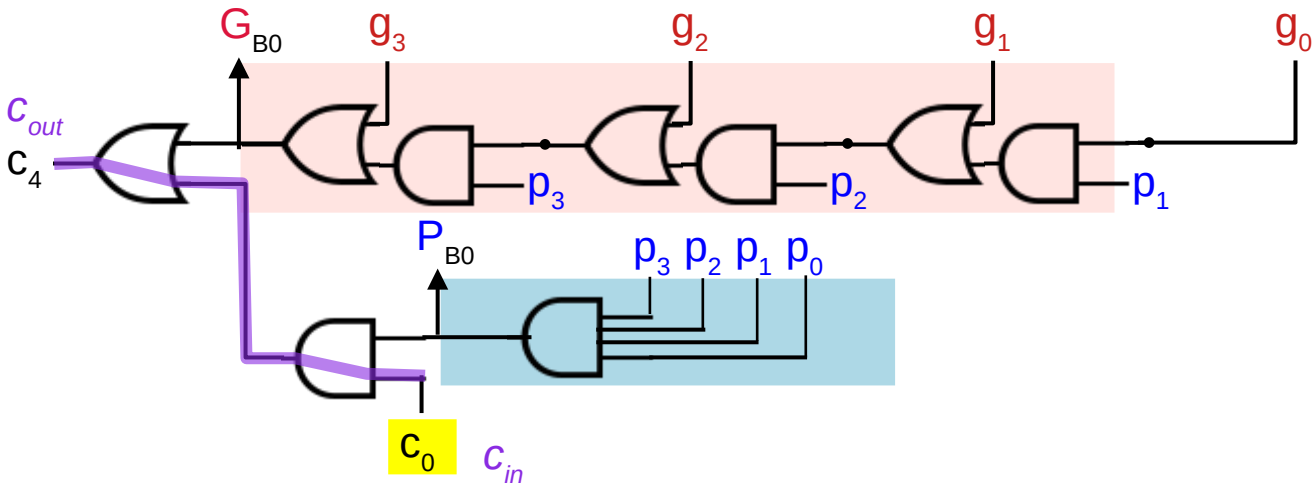
$$P_{B0} = p_3 p_2 p_1 p_0$$

A 4-bit Ripple Carry Adder

the path delay from c_{in} to c_{out} : 8 gate delay



carry paths in the 4 FA's

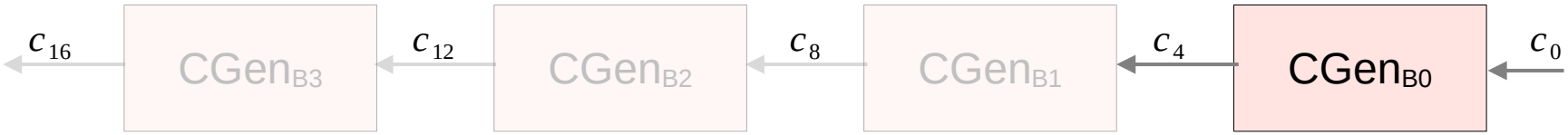


carry lookahead circuits

c_{out}	c_{in}
C_4	C_0
C_8	C_4
C_{12}	C_8
C_{16}	C_{12}

the path delay from c_{in} to c_{out} : 2 gate delay

G_{B0}, P_{B0} : Block Carry Generate and Propagate

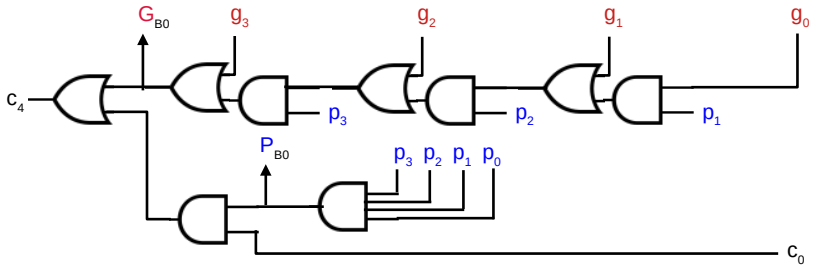


$$c_4 = g_3 + p_3 g_2 + p_3 p_2 g_1 + p_3 p_2 p_1 g_0 + p_3 p_2 p_1 p_0 c_0$$

$$G_{B0} = g_3 + p_3 g_2 + p_3 p_2 g_1 + p_3 p_2 p_1 g_0$$

$$P_{B0} = p_3 p_2 p_1 p_0$$

$$c_4 = G_{B0} + P_{B0} c_0$$



$G_{B0} \leftarrow g_0$: 6Δ
$G_{B0} \leftarrow g_1$: 5Δ
$G_{B0} \leftarrow g_2$: 3Δ
$G_{B0} \leftarrow g_3$: 1Δ

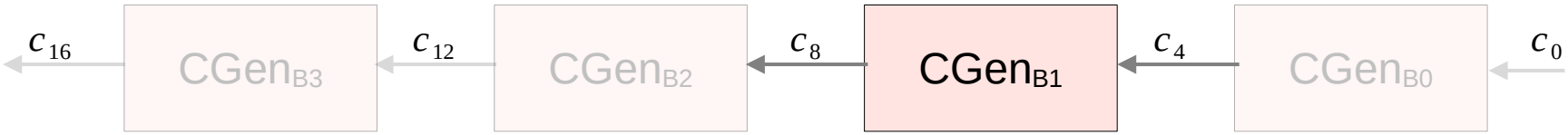
$G_{B0} \leftarrow p_1$: 6Δ
$G_{B0} \leftarrow p_2$: 4Δ
$G_{B0} \leftarrow p_3$: 2Δ

$P_{B0} \leftarrow p_0$: 1Δ
$P_{B0} \leftarrow p_1$: 1Δ
$P_{B0} \leftarrow p_2$: 1Δ
$P_{B0} \leftarrow p_3$: 1Δ

$delay(c_4 \leftarrow g_i) \le 7\Delta$
$delay(c_4 \leftarrow p_i) = 3\Delta$
$delay(c_4 \leftarrow c_0) = 2\Delta$

$delay(G_{B0} \leftarrow g_i) \le 6\Delta$
$delay(G_{B0} \leftarrow p_i) \le 6\Delta$
$delay(P_{B0} \leftarrow p_i) = 1\Delta$

G_{B1}, P_{B1} : Block Carry Generate and Propagate

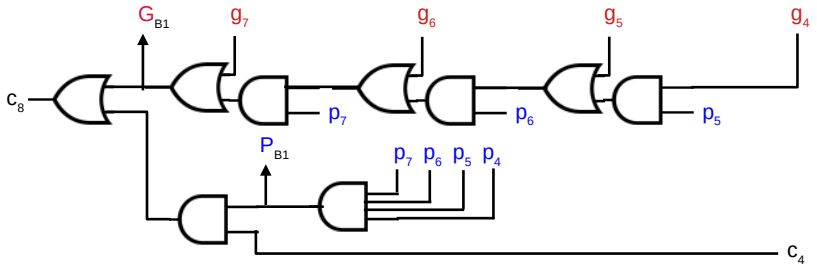


$$c_8 = g_7 + p_7 g_6 + p_7 p_6 g_5 + p_7 p_6 p_5 g_4 + p_7 p_6 p_5 p_4 c_4$$

$$G_{B1} = g_7 + p_7 g_6 + p_7 p_6 g_5 + p_7 p_6 p_5 g_4$$

$$P_{B1} = p_7 p_6 p_5 p_4$$

$$c_8 = G_{B1} + P_{B1} c_4$$



$G_{B1} \leftarrow g_7 : 6\Delta$
$G_{B1} \leftarrow g_6 : 5\Delta$
$G_{B1} \leftarrow g_5 : 3\Delta$
$G_{B1} \leftarrow g_4 : 1\Delta$

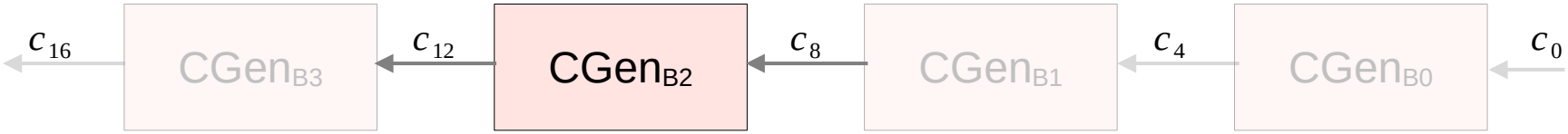
$G_{B1} \leftarrow p_5 : 6\Delta$
$G_{B1} \leftarrow p_6 : 4\Delta$
$G_{B1} \leftarrow p_7 : 2\Delta$

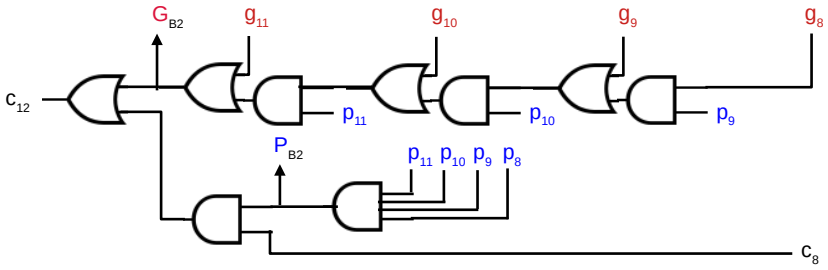
$P_{B1} \leftarrow p_4 : 1\Delta$
$P_{B1} \leftarrow p_5 : 1\Delta$
$P_{B1} \leftarrow p_6 : 1\Delta$
$P_{B1} \leftarrow p_7 : 1\Delta$

$delay(c_8 \leftarrow g_i) \le 7\Delta$
$delay(c_8 \leftarrow p_i) = 3\Delta$
$delay(c_8 \leftarrow c_4) = 2\Delta$

$delay(G_{B1} \leftarrow g_i) \le 6\Delta$
$delay(G_{B1} \leftarrow p_i) \le 6\Delta$
$delay(P_{B1} \leftarrow p_i) = 1\Delta$

G_{B2}, P_{B2} : Block Carry Generate and Propagate





$$c_{12} = g_{11} + p_{11}g_{10} + p_{11}p_{10}g_9 + p_{11}p_{10}p_9g_8 + p_{11}p_{10}p_9p_8c_8$$

$$G_{B2} = g_{11} + p_{11}g_{10} + p_{11}p_{10}g_9 + p_{11}p_{10}p_9g_8$$

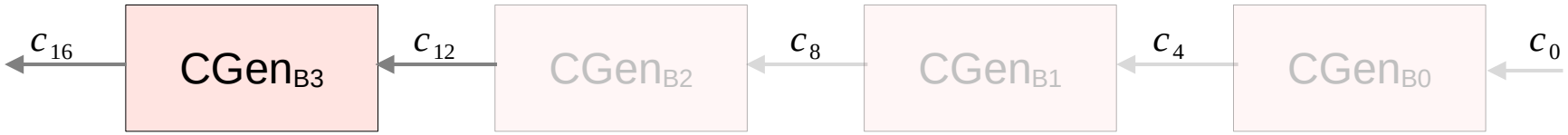
$$P_{B2} = p_{11}p_{10}p_9p_8$$

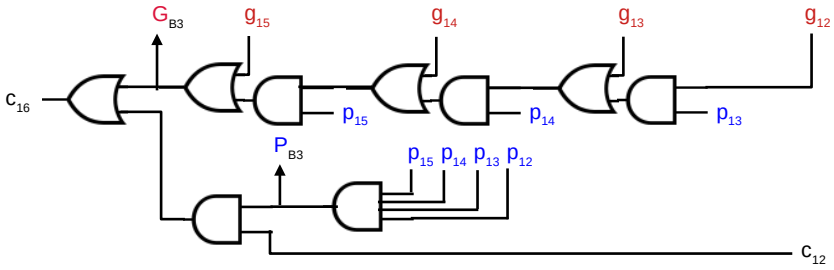
$$c_{12} = G_{B2} + P_{B2}c_8$$

$G_{B2} \leftarrow g_{11}$	: 6Δ			$P_{B2} \leftarrow p_8$	: 1Δ
$G_{B2} \leftarrow g_{10}$	: 5Δ	$G_{B2} \leftarrow p_9$	: 6Δ	$P_{B2} \leftarrow p_9$	: 1Δ
$G_{B2} \leftarrow g_9$	: 3Δ	$G_{B2} \leftarrow p_{10}$	: 4Δ	$P_{B2} \leftarrow p_{10}$	: 1Δ
$G_{B2} \leftarrow g_8$	: 1Δ	$G_{B2} \leftarrow p_{11}$	: 2Δ	$P_{B2} \leftarrow p_{11}$	: 1Δ

$delay(c_{12} \leftarrow g_i)$	$\leq$	7Δ	$delay(G_{B2} \leftarrow g_i)$	$\leq$ 6Δ
$delay(c_{12} \leftarrow p_i)$	$=$	3Δ	$delay(G_{B2} \leftarrow p_i)$	$\leq$ 6Δ
$delay(c_{12} \leftarrow c_8)$	$=$	2Δ	$delay(P_{B2} \leftarrow p_i)$	$=$ 1Δ

G_{B3}, P_{B3} : Block Carry Generate and Propagate





$$c_{16} = g_{15} + p_{15}g_{14} + p_{15}p_{14}g_{13} + p_{15}p_{14}p_{13}g_{12} + p_{15}p_{14}p_{13}p_{12}c_{12}$$

$$G_{B3} = g_{15} + p_{15}g_{14} + p_{15}p_{14}g_{13} + p_{15}p_{14}p_{13}g_{12}$$

$$P_{B3} = p_{15}p_{14}p_{13}p_{12}$$

$$c_{16} = G_{B3} + P_{B3}c_{12}$$

$G_{B3} \leftarrow g_{15}$: 6Δ
$G_{B3} \leftarrow g_{14}$: 5Δ
$G_{B3} \leftarrow g_{13}$: 3Δ
$G_{B3} \leftarrow g_{12}$: 1Δ

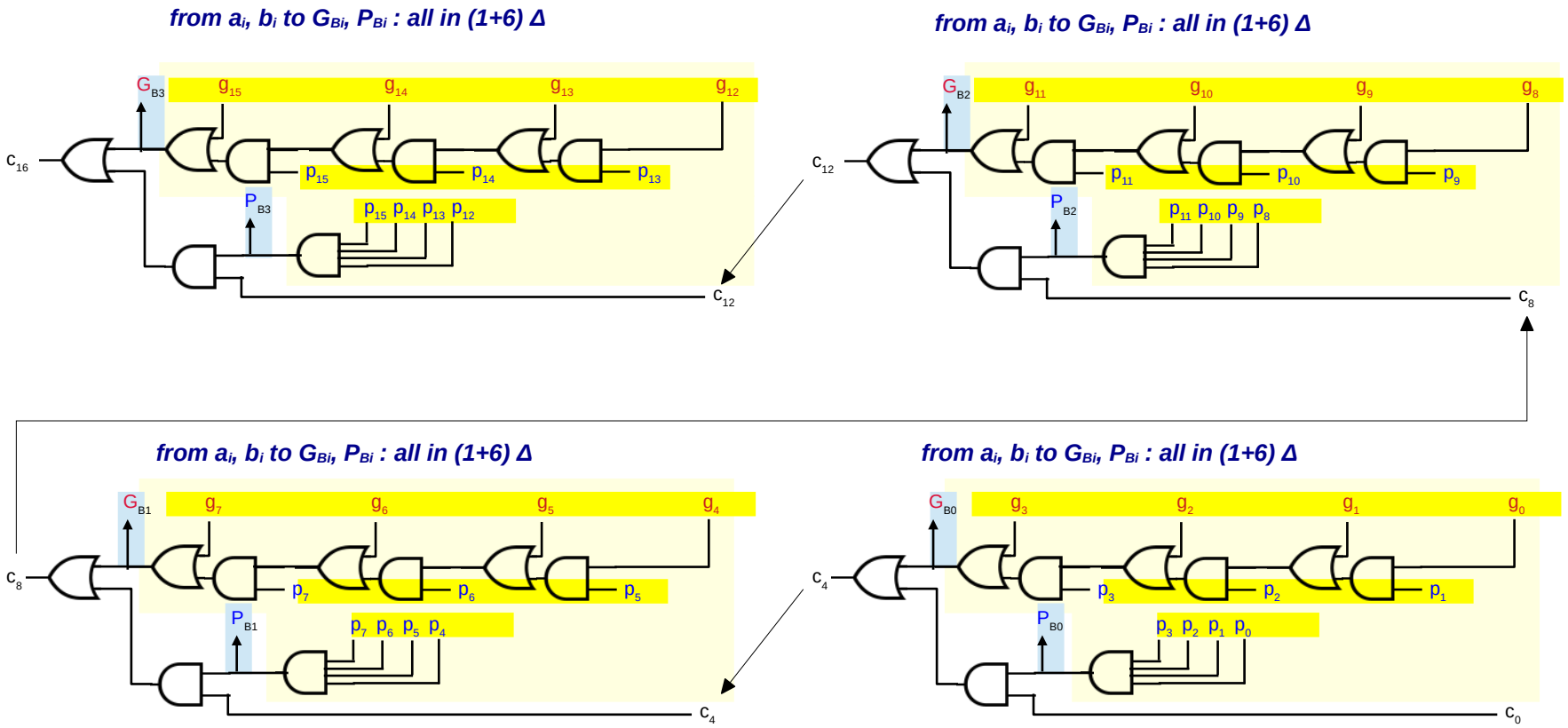
$G_{B3} \leftarrow p_{13}$: 6Δ
$G_{B3} \leftarrow p_{14}$: 4Δ
$G_{B3} \leftarrow p_{15}$: 2Δ

$P_{B3} \leftarrow p_{12}$: 1Δ
$P_{B3} \leftarrow p_{13}$: 1Δ
$P_{B3} \leftarrow p_{14}$: 1Δ
$P_{B3} \leftarrow p_{15}$: 1Δ

$delay(c_{16} \leftarrow g_i) \leq 7\Delta$
$delay(c_{16} \leftarrow p_i) = 3\Delta$
$delay(c_{16} \leftarrow c_{12}) = 2\Delta$

$delay(G_{B3} \leftarrow g_i) \leq 6\Delta$
$delay(G_{B3} \leftarrow p_i) \leq 6\Delta$
$delay(P_{B3} \leftarrow p_i) = 1\Delta$

Delay to generate all G_{Bi} and P_{Bi}



$$delay(p_i, g_i \leftarrow a_i, b_i) = 1\Delta$$
$$delay(G_{Bi} \leftarrow p_i, g_i) \le 6\Delta$$
$$delay(P_{Bi} \leftarrow p_i, g_i) = 1\Delta$$

$$delay(c_o \leftarrow G_{Bi}) = 1\Delta$$
$$delay(c_o \leftarrow P_{Bi}) = 2\Delta$$
$$delay(c_o \leftarrow c_i) = 2\Delta$$

$$delay(c_o \leftarrow a_i, b_i) \le 8\Delta$$
$$delay(c_o \leftarrow a_i, b_i) = 4\Delta$$
$$delay(c_o \leftarrow c_i) = 2\Delta$$

p_i, g_i : all computed in parallel

T_{pg}, T_{pg_block}

T_{pg} : path delay from a_i, b_i to p_i, g_i — 1 Δ

T_{pg_block} : path delay from p_i, g_i to G_{Bi}, P_{Bi} — 6 Δ

$T_{pg} + T_{pg\_block}$: path delay from a_i, b_i to G_{Bi}, P_{Bi} — 7 Δ

after $T_{pg} + T_{pg\_block}$, all G_{bi}'s , P_{bi}'s are available.

But each carry lookahead circuit can compute
its own c_{out}, only after c_{in} is ready.

Thus, $c_0 \rightarrow c_4 \rightarrow c_8 \rightarrow c_{12}$ forms a critical path

in the 1st block, the path delay $c_0 \rightarrow c_4$ takes T_{AND_OR}
in the 2nd block, the path delay $c_4 \rightarrow c_8$ takes T_{AND_OR}
in the 3rd block, the path delay $c_8 \rightarrow c_{12}$ takes T_{AND_OR}
in the 4th block, the path delay $c_{12} \rightarrow c_{16}$ takes T_{AND_OR}

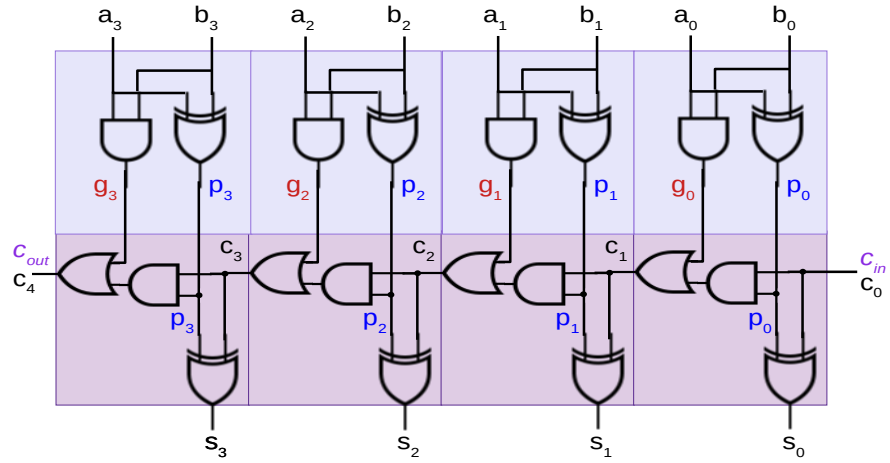
T_{pg}, T_{pg_block}

	$c_{in} \rightarrow c_{out}$	Path delay	
in the 1st block,	$c_0 \rightarrow c_4$	T_{AND_OR}	$= 2\,\Delta$
in the 2nd block,	$c_4 \rightarrow c_8$	T_{AND_OR}	$= 2\,\Delta$
in the 3rd block,	$c_8 \rightarrow c_{12}$	T_{AND_OR}	$= 2\,\Delta$
in the 4th block,	$c_{12} \rightarrow c_{16}$	T_{AND_OR}	$= 2\,\Delta$

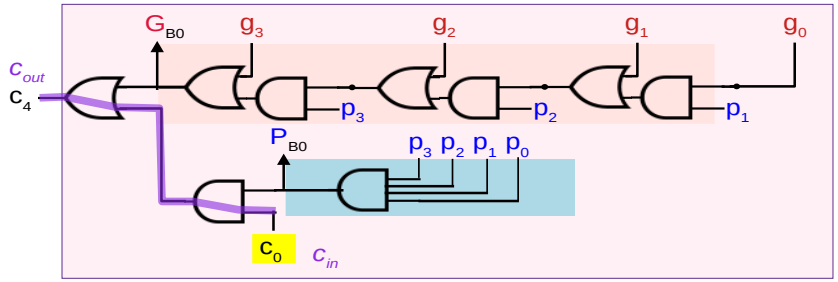
	$c_o \rightarrow c_{out}$	Path delay	
in the 1st block,	$c_0 \rightarrow c_4$	T_{AND_OR}	$= 2\,\Delta$
in the 2nd block,	$c_0 \rightarrow c_8$	T_{AND_OR}	$= 4\,\Delta$
in the 3rd block,	$c_0 \rightarrow c_{12}$	T_{AND_OR}	$= 6\,\Delta$
in the 4th block,	$c_0 \rightarrow c_{16}$	T_{AND_OR}	$= 8\,\Delta$

A 4-bit Ripple Carry Adder

While the carry lookahead operation is performed,
the sum bits are computed by a ripple carry adder



k = 4 Full Adders → a 4-bit ripple carry adder



carry lookahead circuit

T_{pg}, T_{pg_block}

While carry lookahead operation is performed,
the sum bits are computed by a ripple carry adder

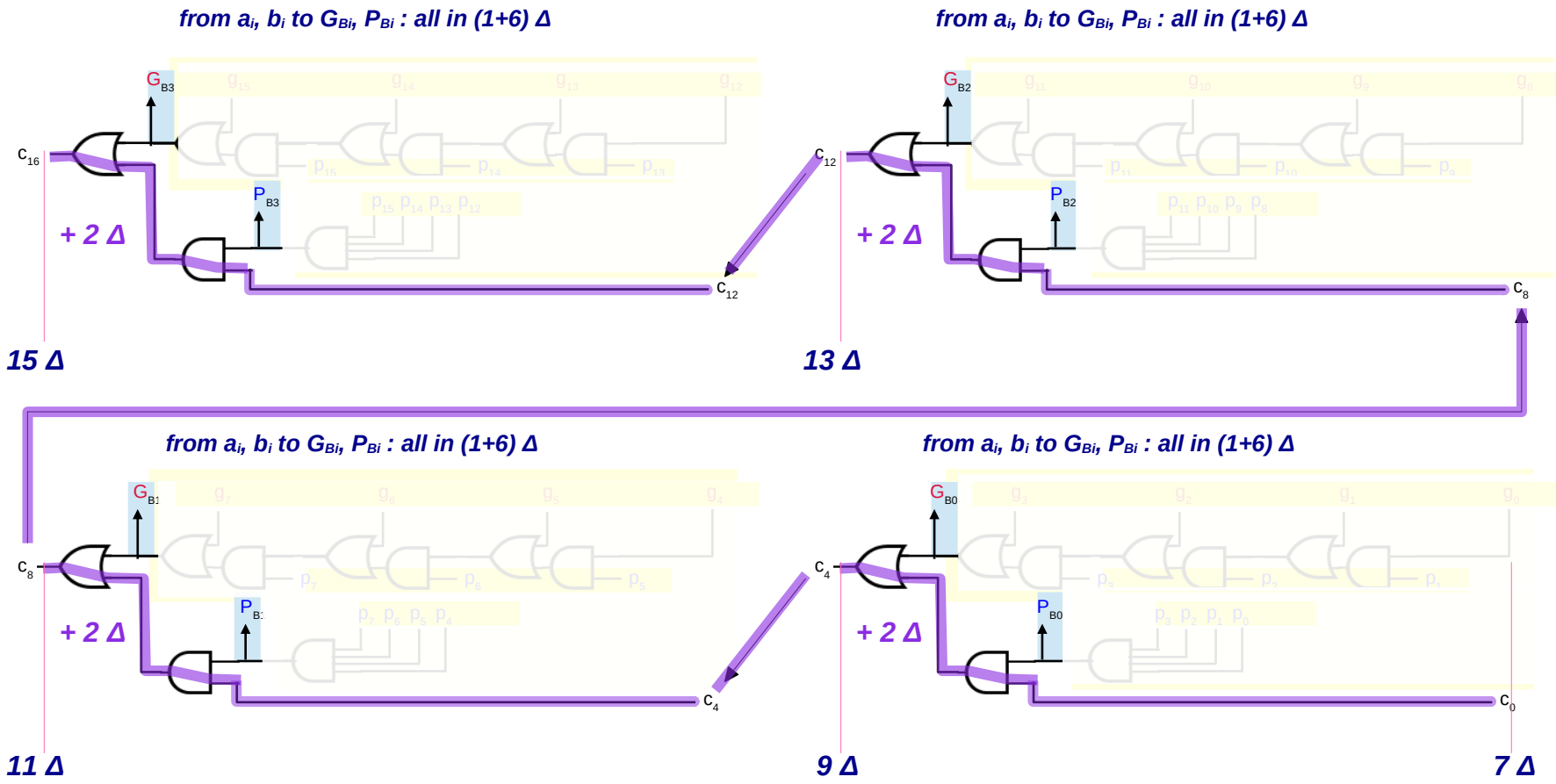
the 1st block's ripple carry adder
starts to computes its sum bits using c_0 *after* $T_{pg} + T_{pg\_block}$

the 2nd block's ripple carry adder
starts to computes its sum bits using c_4 *after* $T_{pg} + T_{pg\_block} + 1 \cdot T_{AND\_OR}$

the 3rd block's ripple carry adder
starts to computes its sum bits using c_8 *after* $T_{pg} + T_{pg\_block} + 2 \cdot T_{AND\_OR}$

the 4th block's ripple carry adder
starts to computes its sum bits using c_{12} *after* $T_{pg} + T_{pg\_block} + 3 \cdot T_{AND\_OR}$

Critical paths after 7 Δ



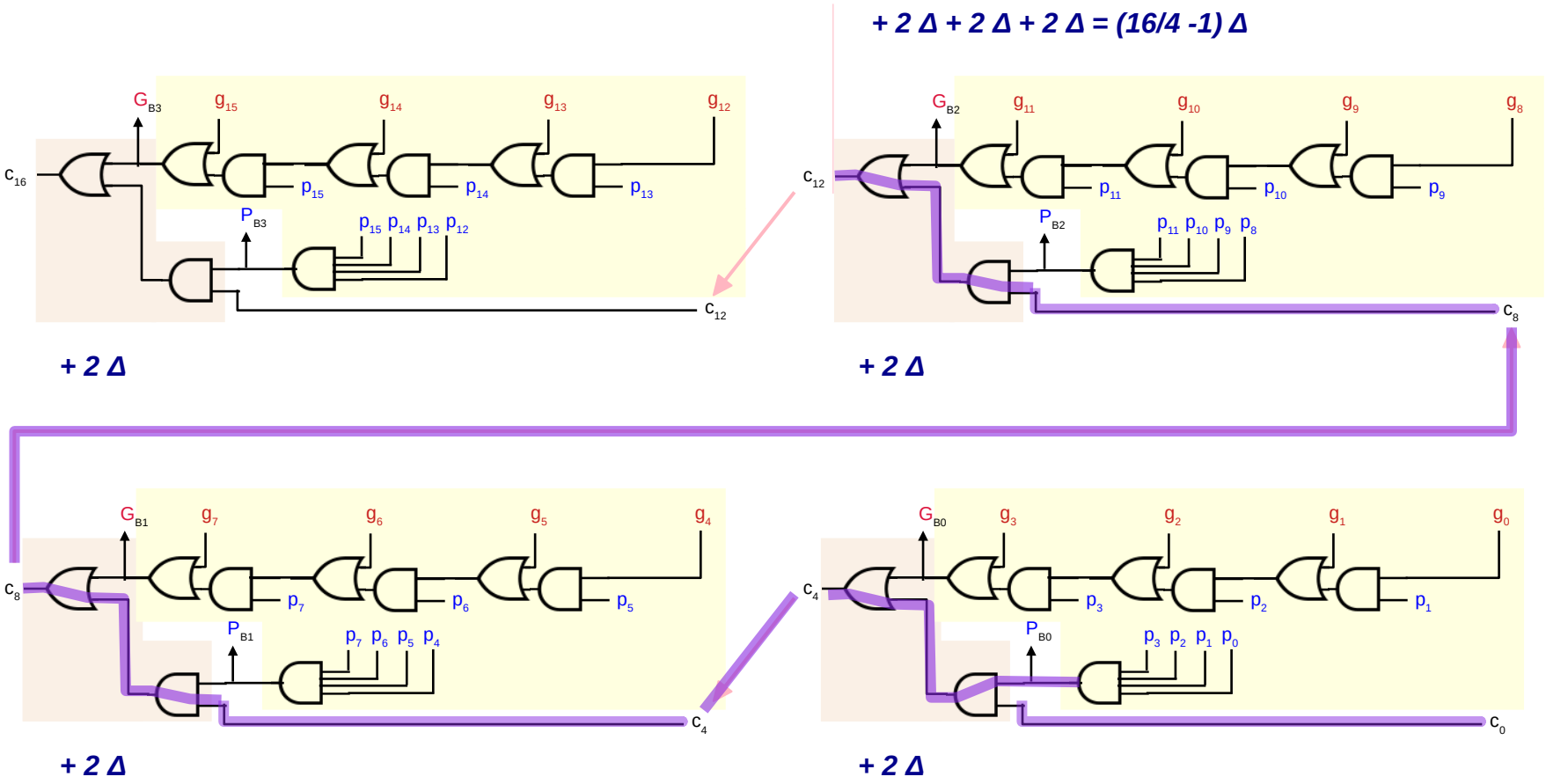
$$delay(p_i, g_i \leftarrow a_i, b_i) = 1\Delta$$
$$delay(G_{Bi} \leftarrow p_i, g_i) \le 6\Delta$$
$$delay(P_{Bi} \leftarrow p_i, g_i) = 1\Delta$$

$$delay(c_o \leftarrow G_{Bi}) = 1\Delta$$
$$delay(c_o \leftarrow P_{Bi}) = 2\Delta$$
$$delay(c_o \leftarrow c_i) = 2\Delta$$

$$delay(c_o \leftarrow a_i, b_i) \le 8\Delta$$
$$delay(c_o \leftarrow a_i, b_i) = 4\Delta$$
$$delay(c_o \leftarrow c_i) = 2\Delta$$

p_i, g_i : all computed in parallel

T_{AND_OR} *(N/k-1)

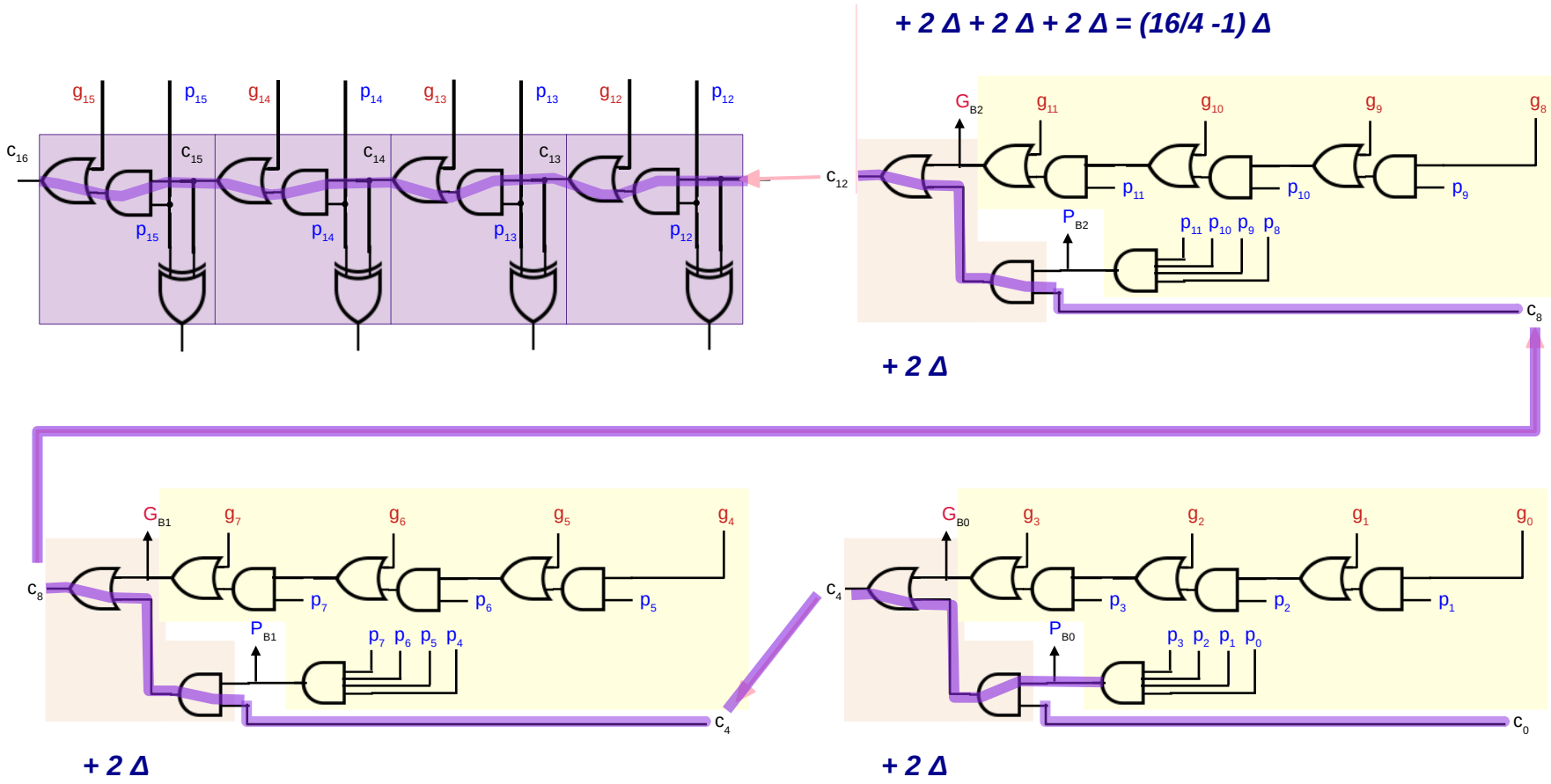


$$delay(c_{4(i+1)} \leftarrow G_{Bi}) = 1\Delta$$
$$delay(c_{4(i+1)} \leftarrow B_{Bi}) = 2\Delta$$
$$delay(c_{4(i+1)} \leftarrow c_{4i}) = 2\Delta$$

$$delay(G_{Bi} \leftarrow a_i, b_i) \leq 7\Delta$$
$$delay(P_{Bi} \leftarrow a_i, b_i) = 2\Delta$$

$$delay(c_{4(i+1)} \leftarrow a_i, b_i) \leq 8\Delta$$
$$delay(c_{4(i+1)} \leftarrow a_i, b_i) = 4\Delta$$
$$delay(c_{4(i+1)} \leftarrow c_{4i}) = 2\Delta$$

computed in serial



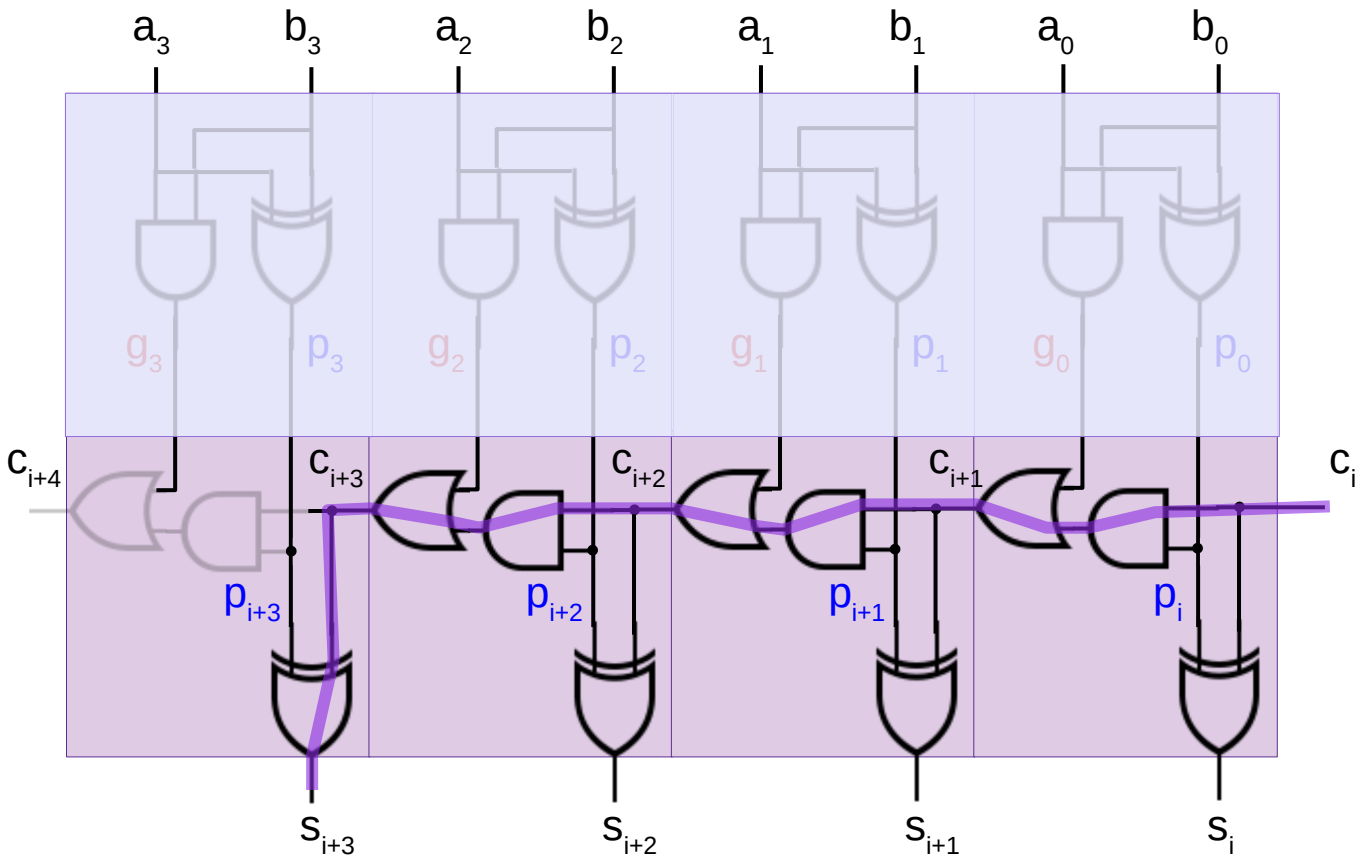
$$delay(c_{4(i+1)} \leftarrow G_{Bi}) = 1\Delta$$
$$delay(c_{4(i+1)} \leftarrow B_{Bi}) = 2\Delta$$
$$delay(c_{4(i+1)} \leftarrow c_{4i}) = 2\Delta$$

$$delay(G_{Bi} \leftarrow a_i, b_i) \le 7\Delta$$
$$delay(P_{Bi} \leftarrow a_i, b_i) = 2\Delta$$

$$delay(c_{4(i+1)} \leftarrow a_i, b_i) \le 8\Delta$$
$$delay(c_{4(i+1)} \leftarrow a_i, b_i) = 4\Delta$$
$$delay(c_{4(i+1)} \leftarrow c_{4i}) = 2\Delta$$

computed in serial

A 4-bit Ripple Carry Adder



Lookahead circuits compute c_{i+4}

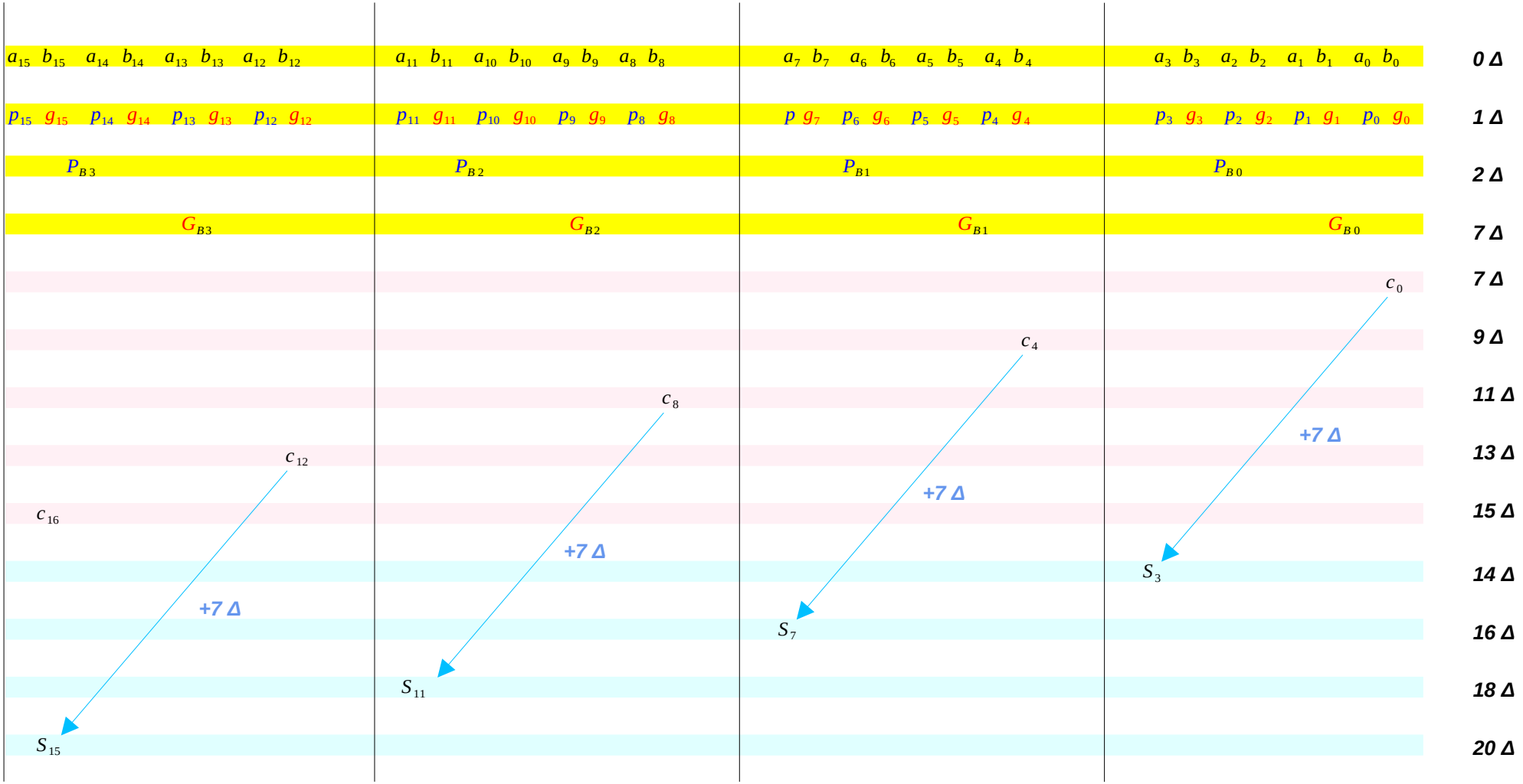
When c_i is available, all g_i's and p_i's are already available

$$delay(S_{i+3} \leftarrow c_i) = 7\Delta$$

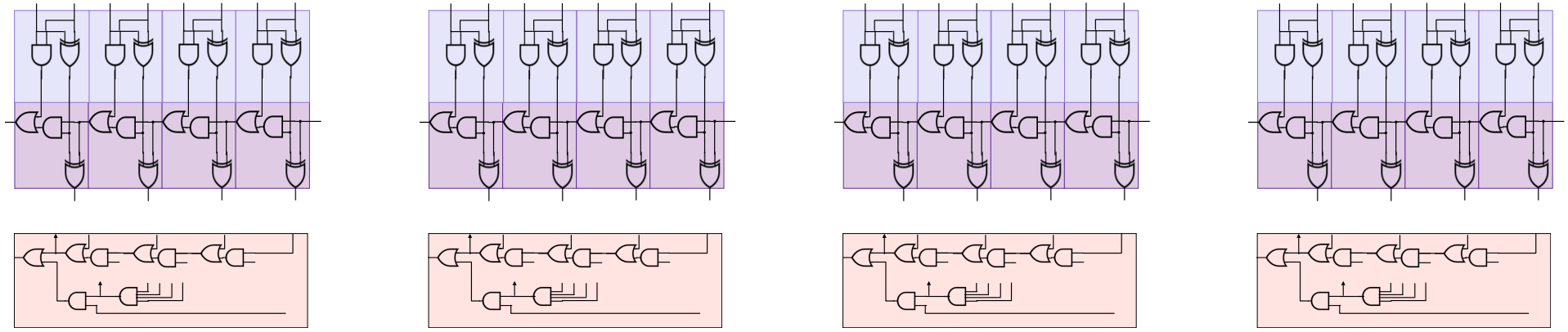
Critical Path : c_i – c_{i+1} – c_{i+2} – c_{i+3} – S_{i+3} : 7 gate delay

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

P_{Bi}'s, G_{Bi}'s are computed in parallel

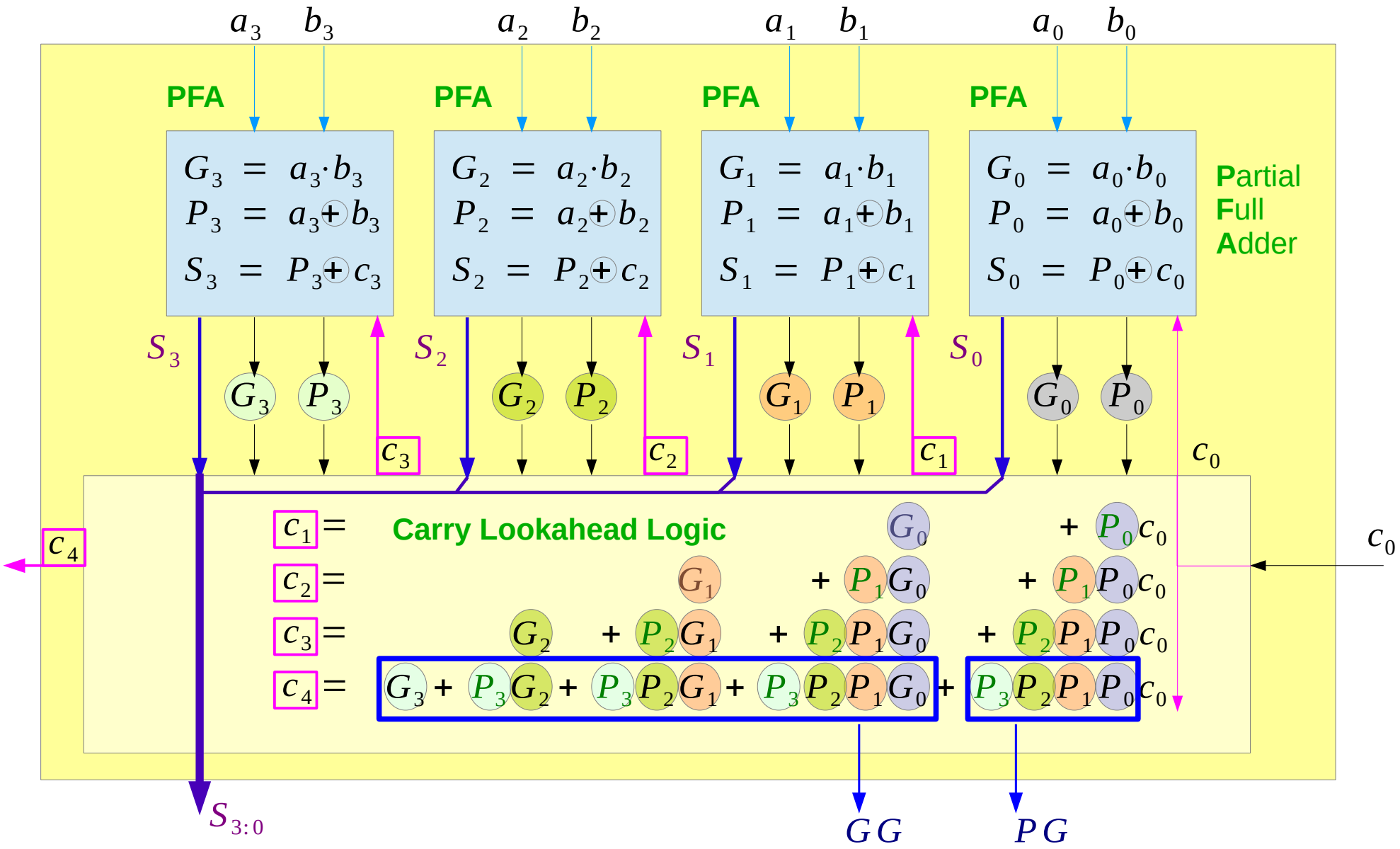


A 4-bit Ripple Carry Adder

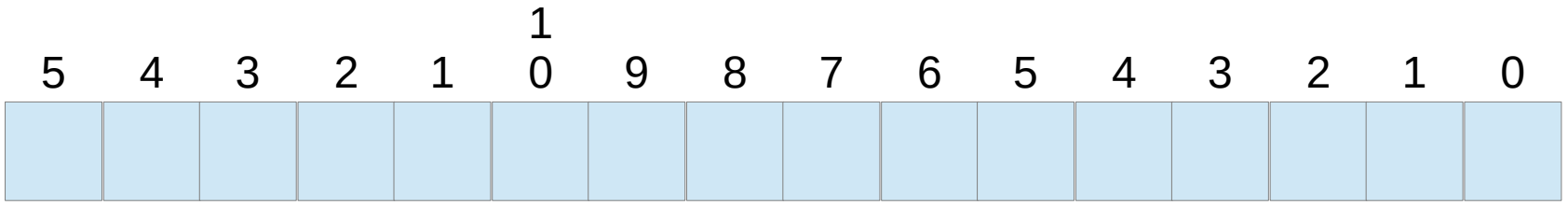


https://upload.wikimedia.org/wikiversity/en/1/18/RCA.Note.H.1.20151215.pdf

4-bit CLA

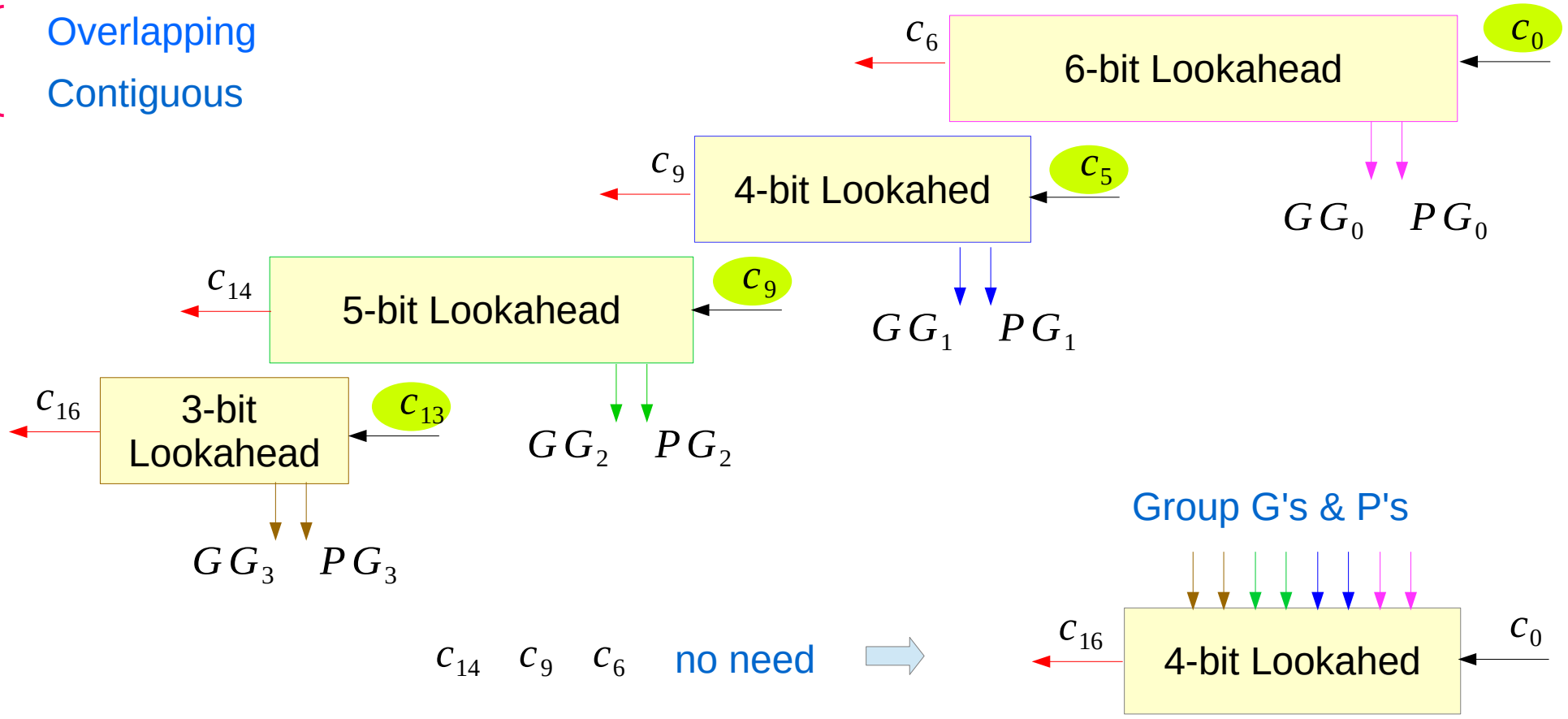

a_3 b_3 a_2 b_2 a_1 b_1 a_0 b_0
PFA
PFA
PFA
PFA
$G_3 = a_3 \cdot b_3$
$P_3 = a_3 \oplus b_3$
$S_3 = P_3 \oplus c_3$
$G_2 = a_2 \cdot b_2$
$P_2 = a_2 \oplus b_2$
$S_2 = P_2 \oplus c_2$
$G_1 = a_1 \cdot b_1$
$P_1 = a_1 \oplus b_1$
$S_1 = P_1 \oplus c_1$
$G_0 = a_0 \cdot b_0$
$P_0 = a_0 \oplus b_0$
$S_0 = P_0 \oplus c_0$
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

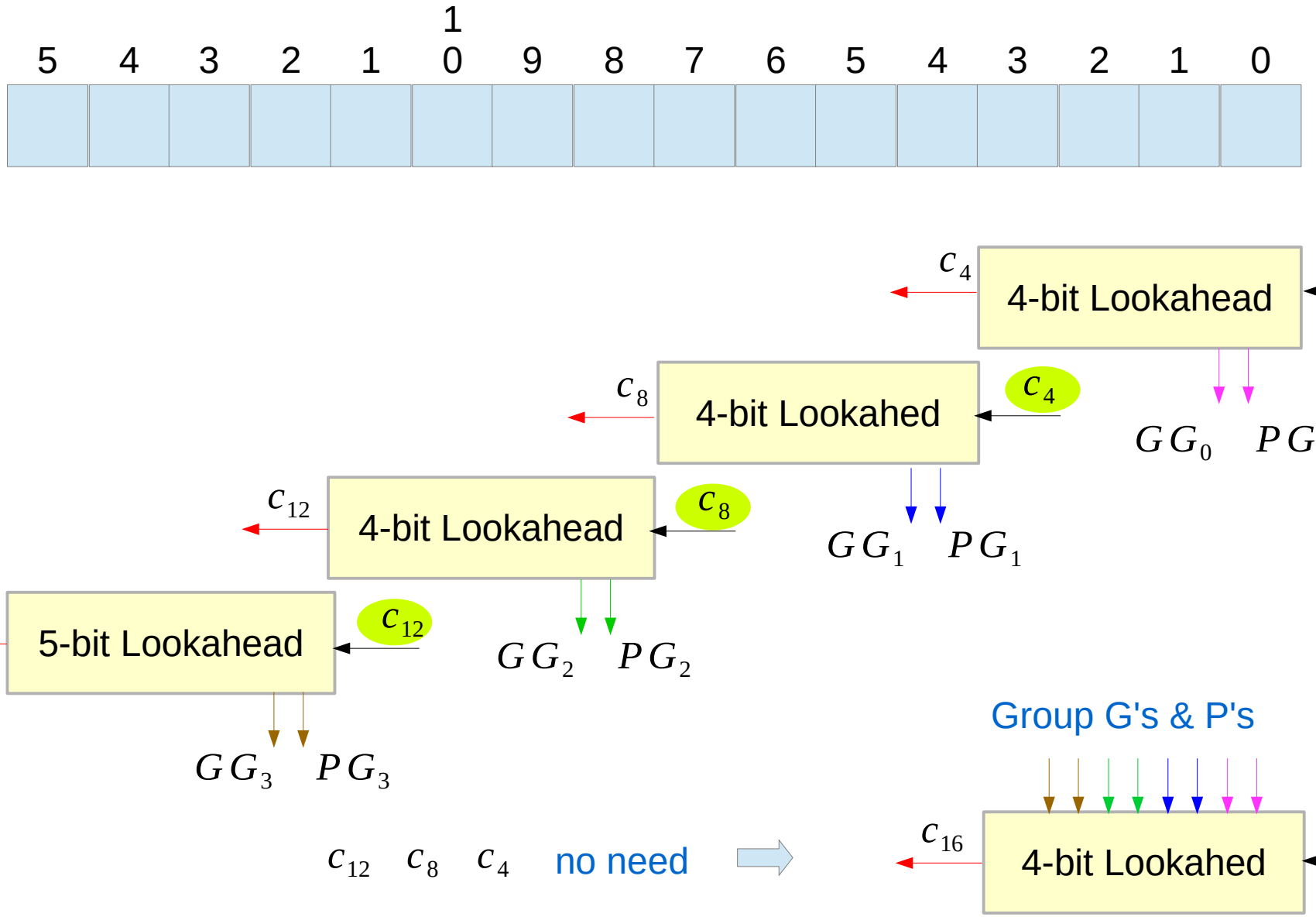


Overlapping

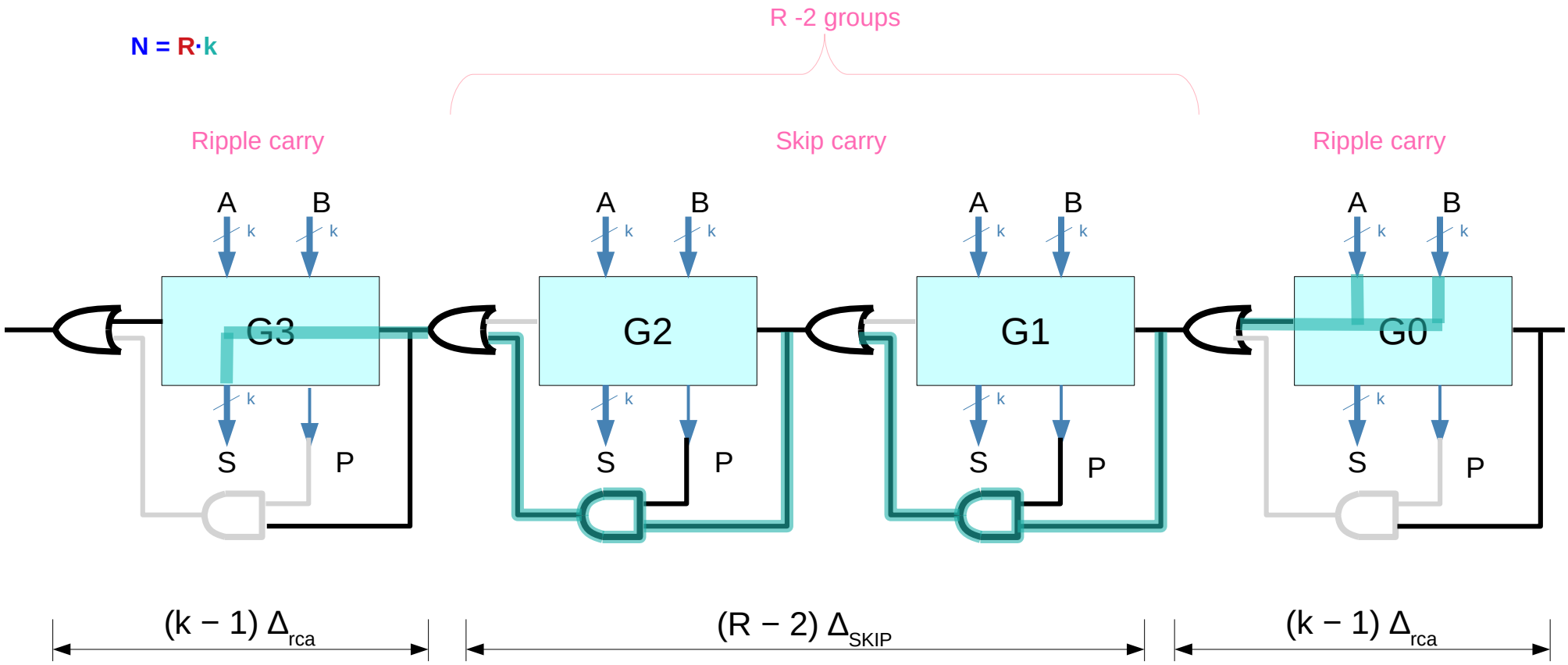
Contiguous



Contiguous Multi-level Carry Lookahead



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