Carry Skip Adder (5A)

-
-

Young W. Lim
11/12/24

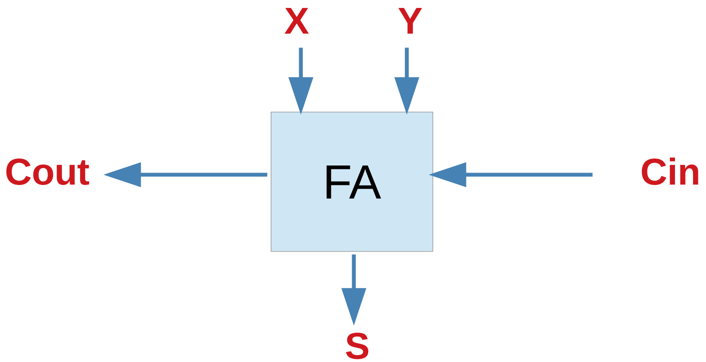
https://en.wikipedia.org/wiki/AND_gate
https://en.wikipedia.org/wiki/OR_gate
https://en.wikipedia.org/wiki/XOR_gate
https://en.wikipedia.org/wiki/NAND_gate

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

This document was produced by using OpenOffice and Octave.

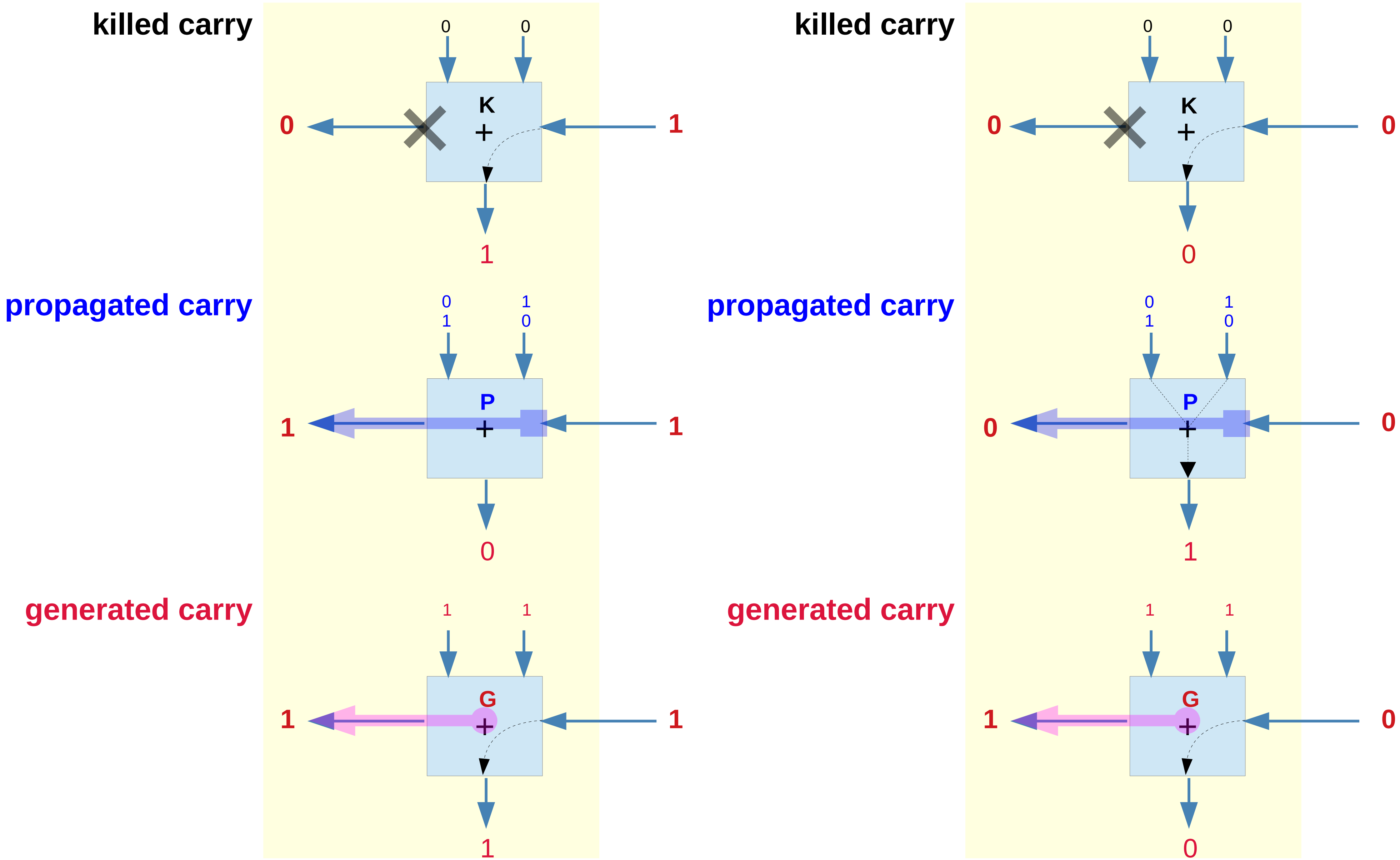
Carry Kill, Propagate, Generate conditions (1)

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



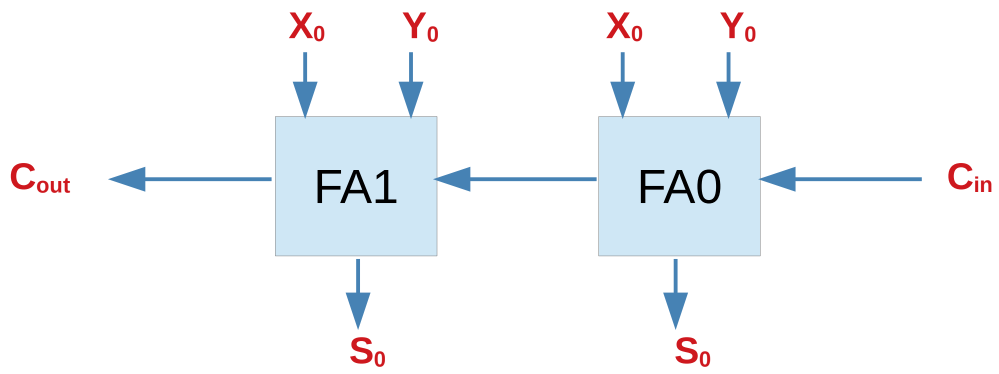
https:://electronics.stackexchange.com/questions/21251/critical-path-for-carry-skip-adder

Carry Kill, Propagate, Generate conditions (2)



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

X	Y		
0	0	K	Kill (=$\overline{PG}$)
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

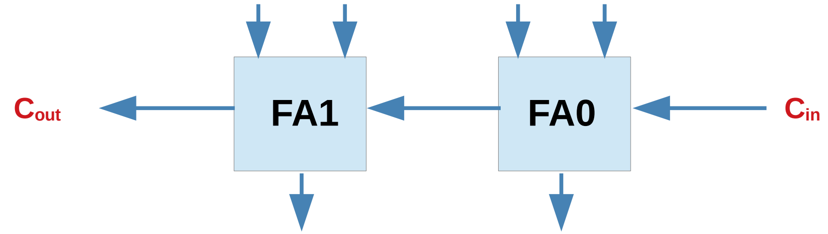
Critical path – all FA's in propagate mode

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

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

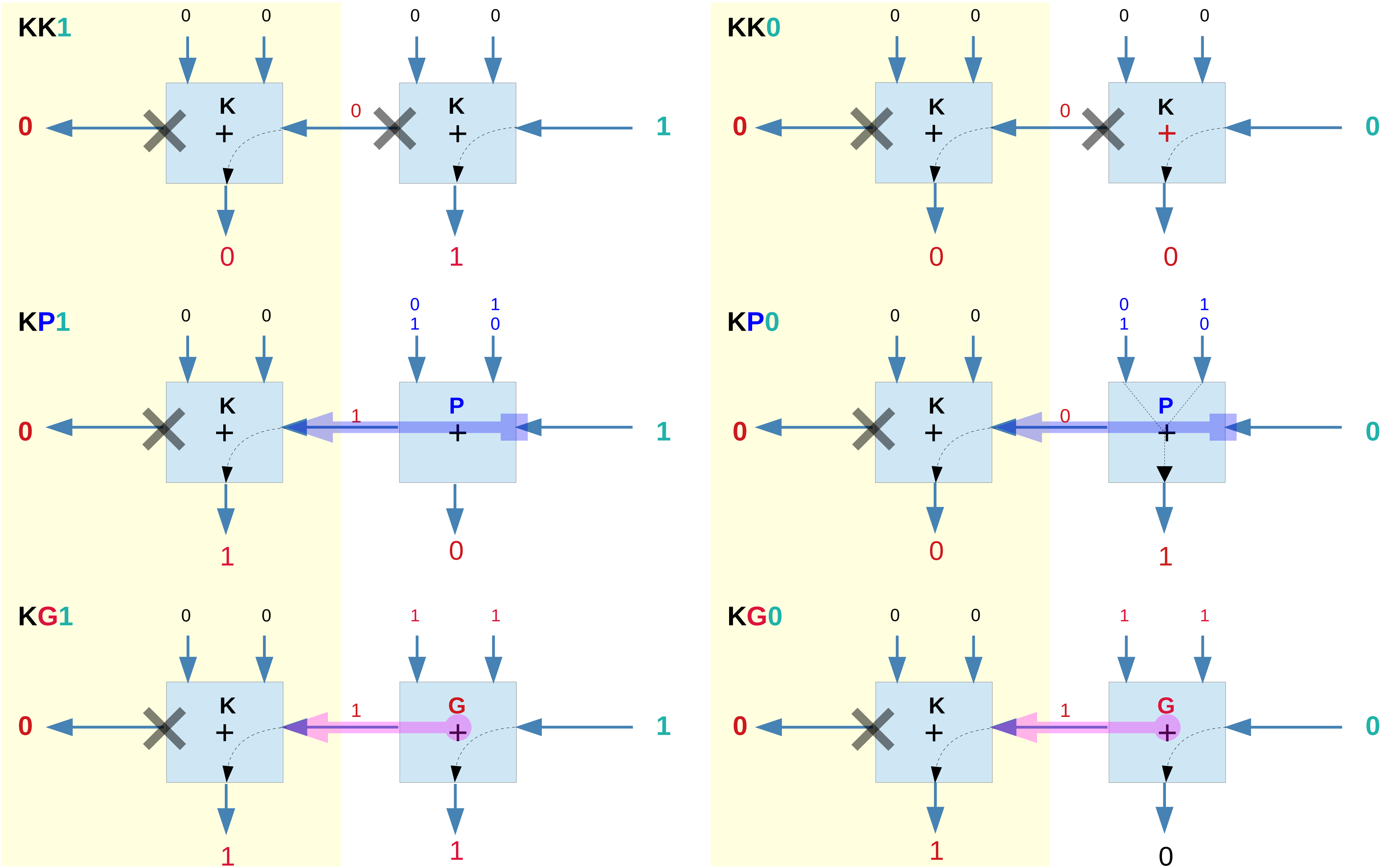
K, **P**, and **G** conditions in a 2-bit adder (2)

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

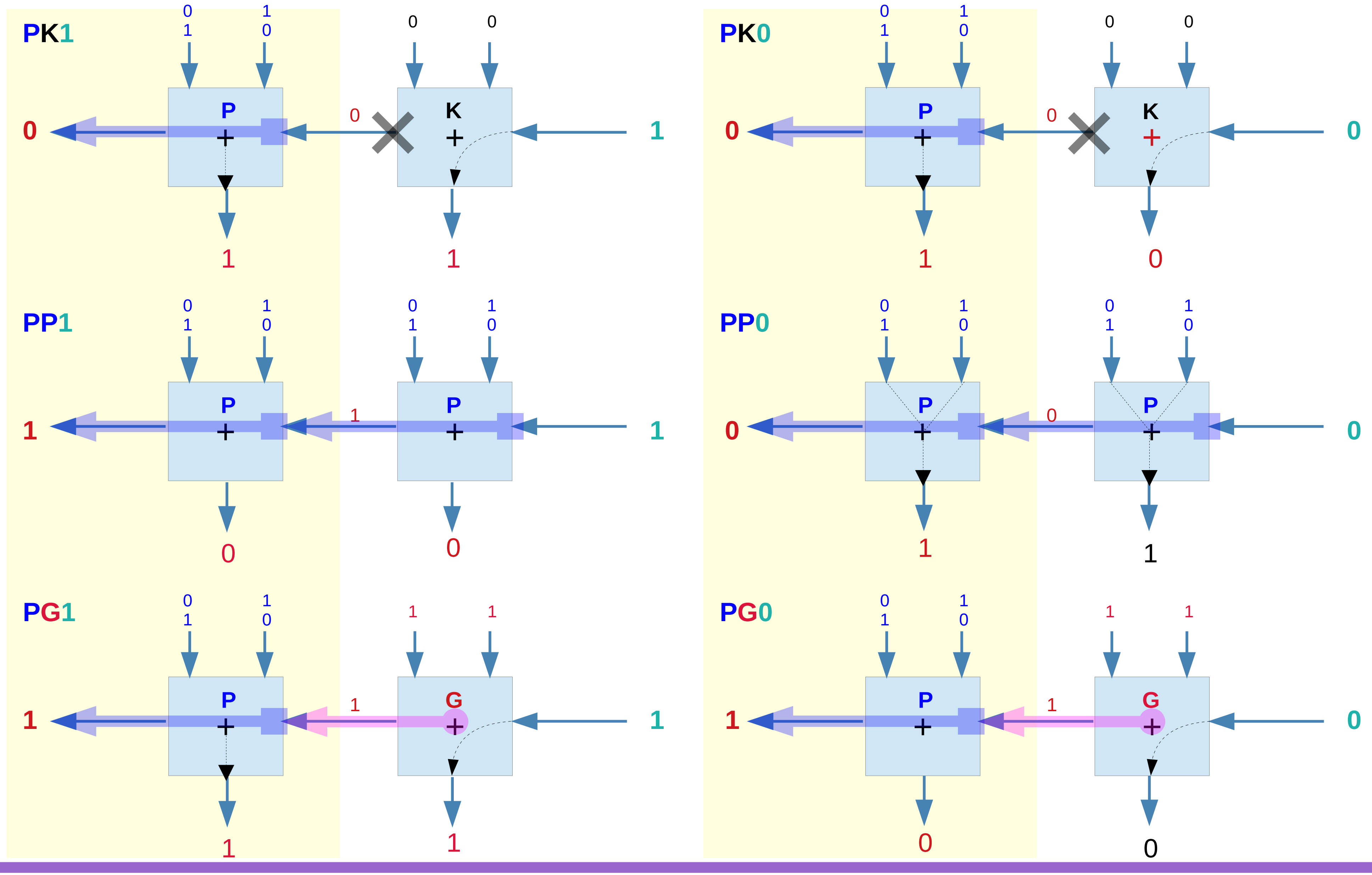


K	**K**	**0**
K	**K**	**1**
K	**P**	**0**
K	**P**	**1**
K	**G**	**0**
K	**G**	**1**
P	**K**	**0**
P	**K**	**1**
P	**P**	**0**
P	**P**	**1**
P	**G**	**0**
P	**G**	**1**
G	**K**	**0**
G	**K**	**1**
G	**P**	**0**
G	**P**	**1**
G	**G**	**0**
G	**G**	**1**

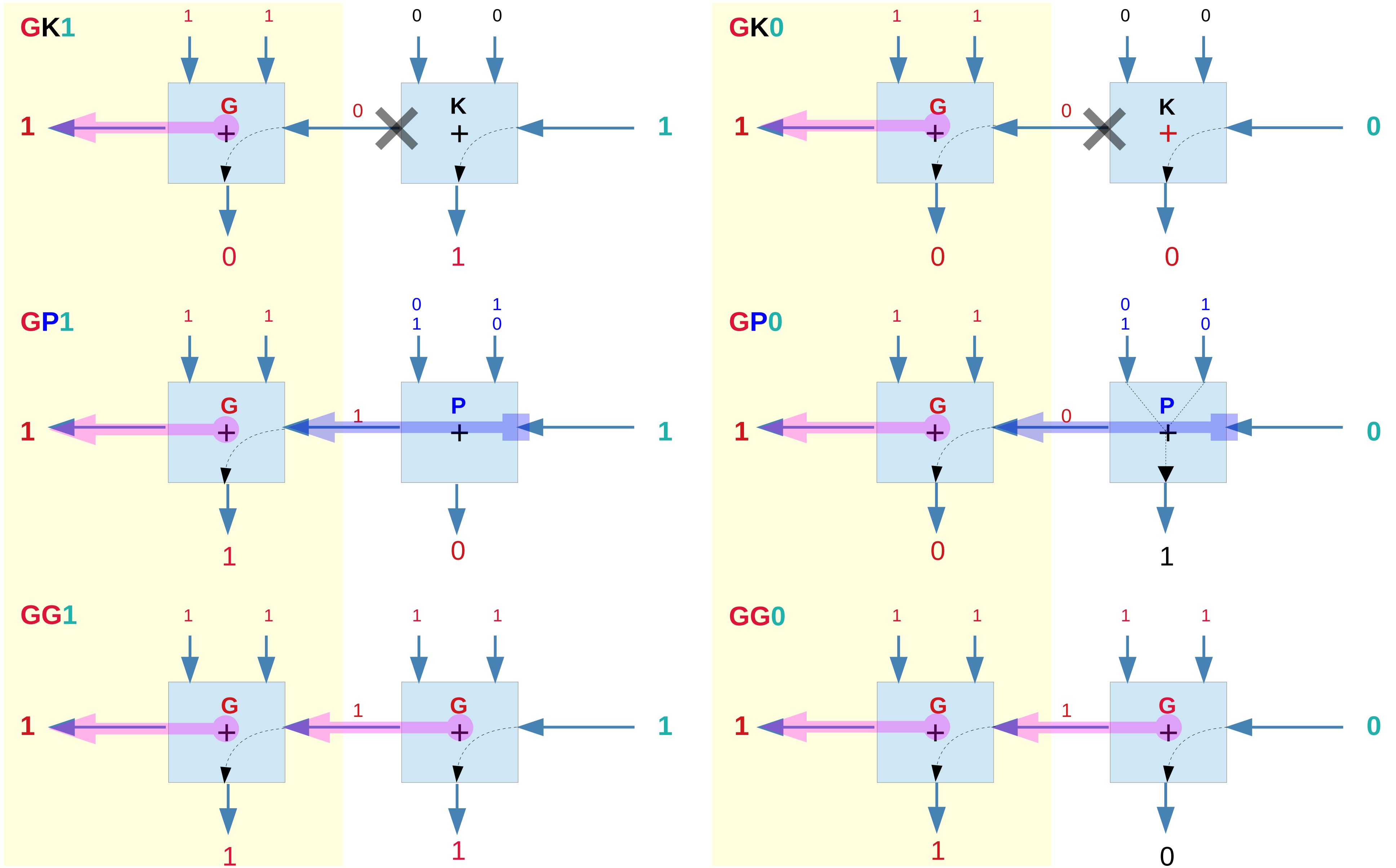
1. Cases when **FA1** is in the **K** mode



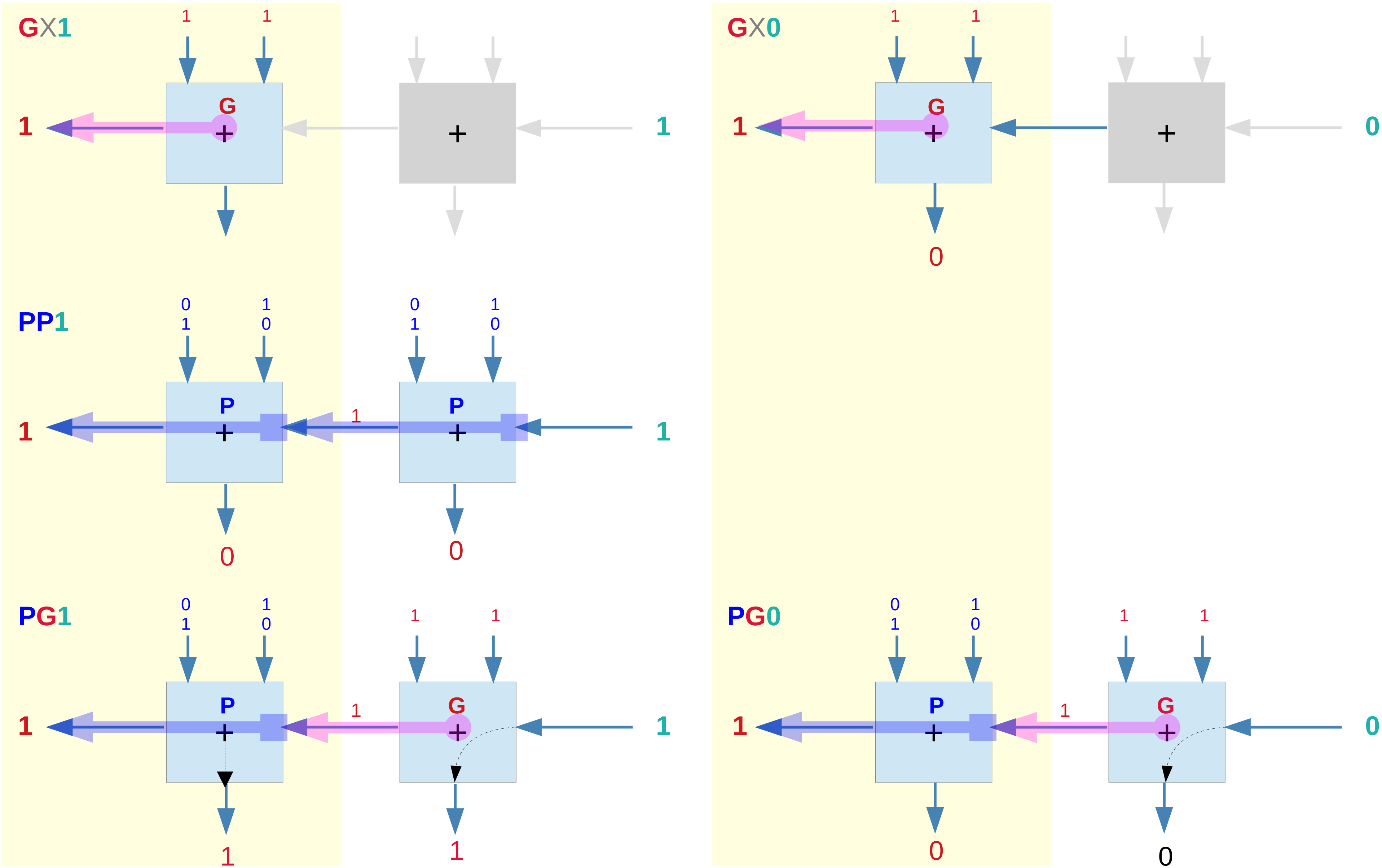
2. Cases when **FA1** is in the **P** mode



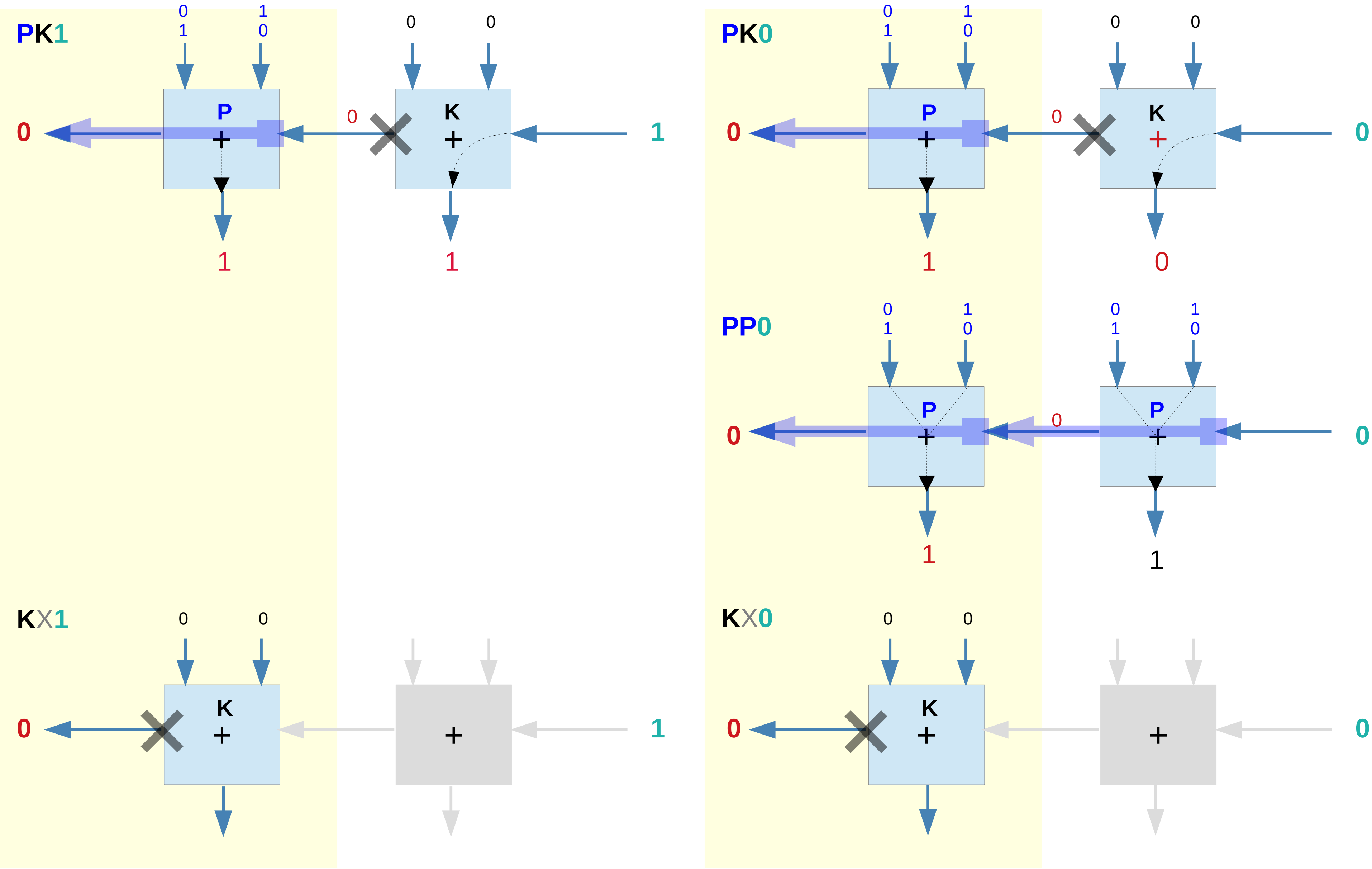
3. Cases when **FA1** is in the **G** mode



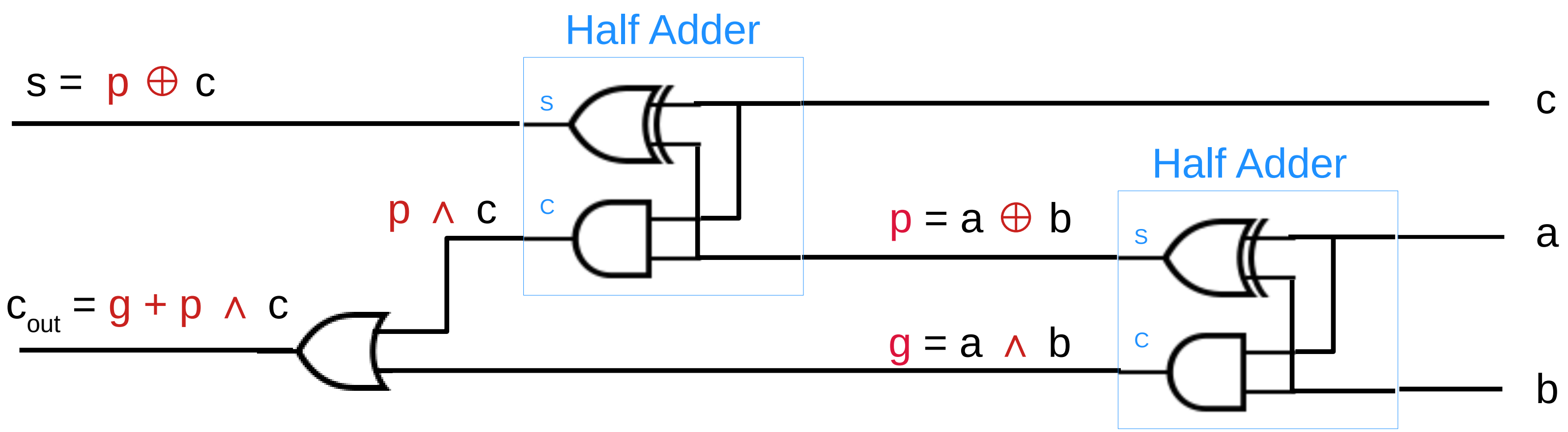
Cases for C_{out} = 1



Cases for C_{out} = 0



FA with P & G

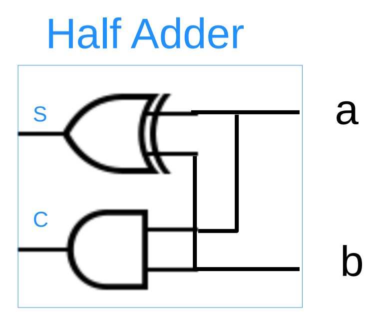


Half Adder
$S = a \oplus b$
$C = a \wedge b$

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

p $S = a \oplus b$

g $C = a \wedge b$



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

Full adder with additional generate and propagate signals.

Ripple Carry Adder

$p_i = a_i \oplus b_i$

$g_i = a_i \wedge b_i$

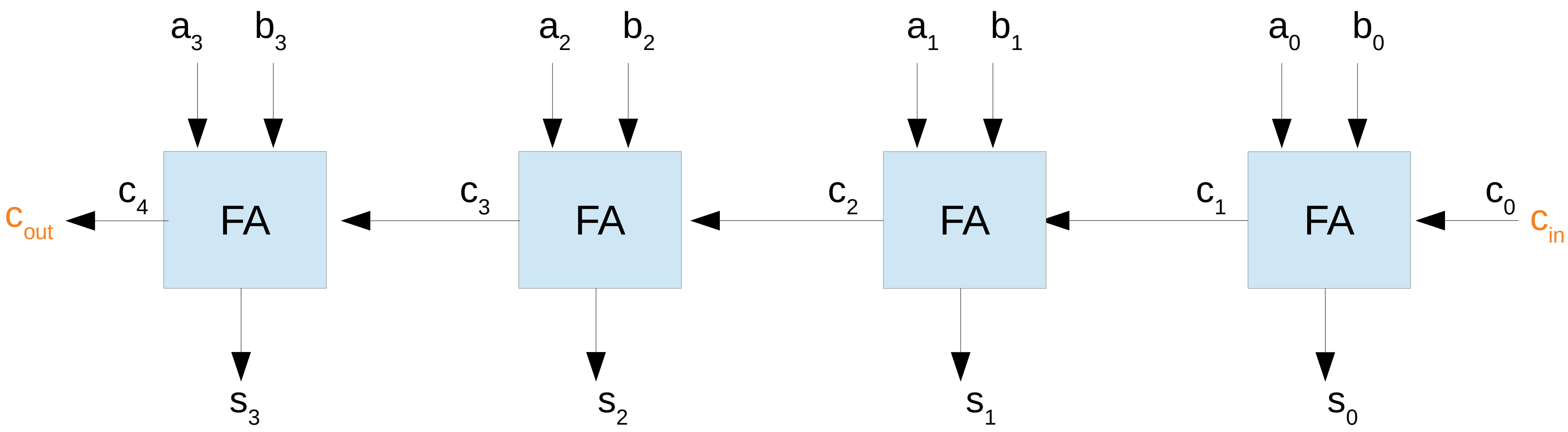
$c_1 = g_0 + p_0 \wedge c_0$

$c_2 = g_1 + p_1 \wedge c_1$

$c_3 = g_2 + p_2 \wedge c_2$

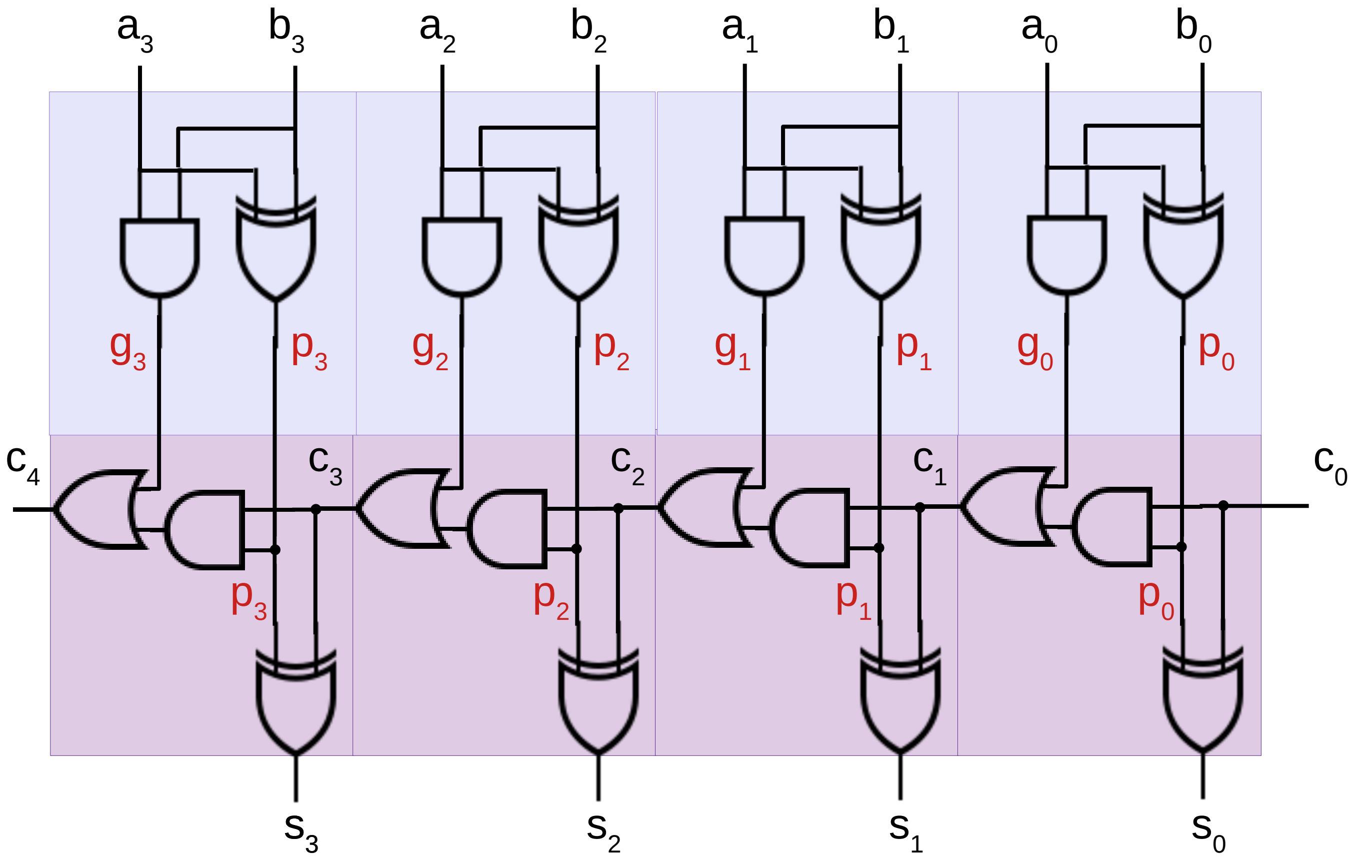
$c_4 = g_3 + p_3 \wedge c_3$

generated carry (the g terms) — propagated carry (the $p \wedge c$ terms)



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

4-bit Full Adder with P and G



Half Adder

$p_i = a_i \oplus b_i$

$g_i = a_i \wedge b_i$

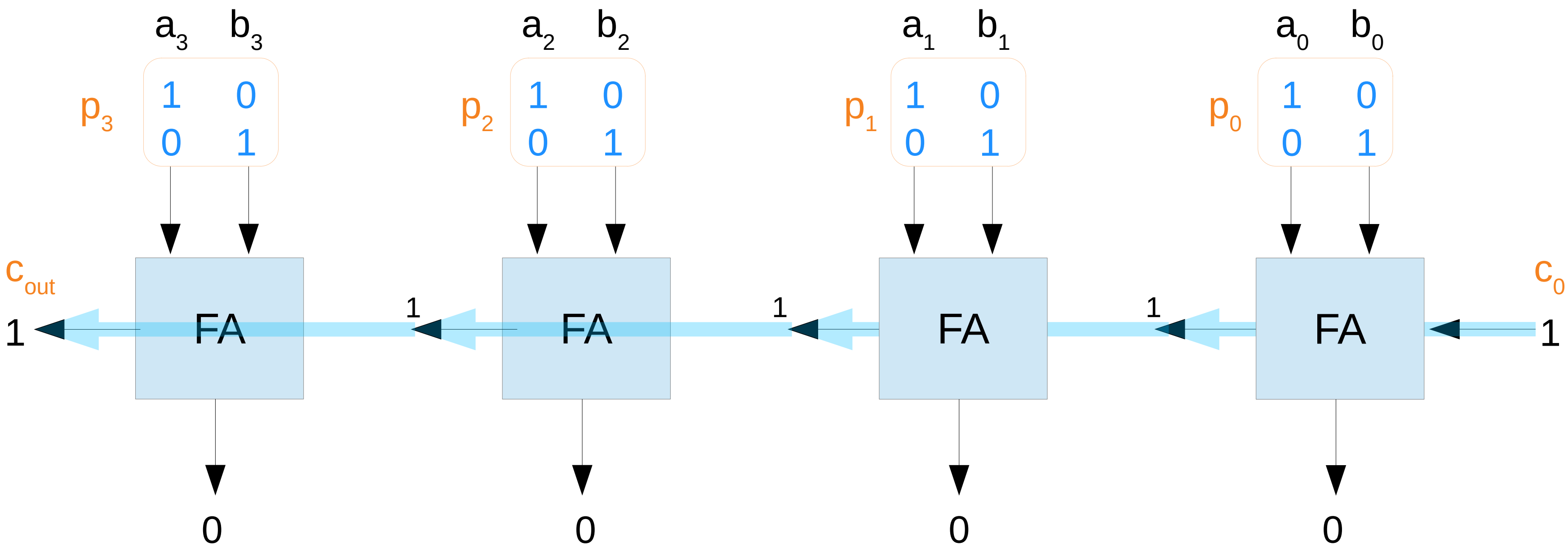
Half Adder

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

$s_i = p_i \oplus c_i$

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

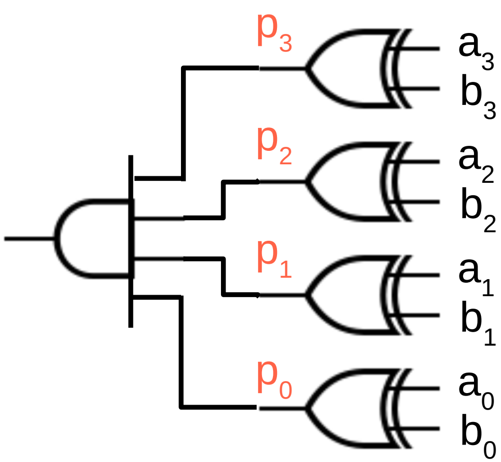
C_0 propagation condition



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

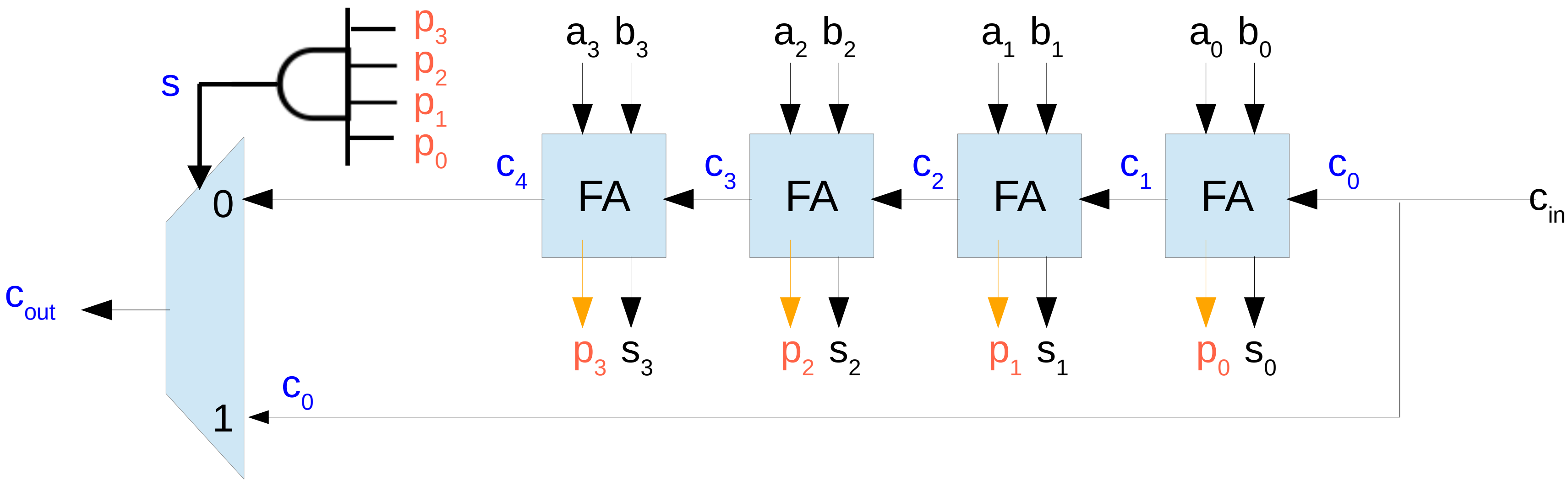
$$
\begin{aligned}
s &= p_3 \wedge p_2 \wedge p_1 \wedge p_0 = p_{[3:0]} \\
&= (a_3 \oplus b_3) \\
&\wedge (a_2 \oplus b_2) \\
&\wedge (a_1 \oplus b_1) \\
&\wedge (a_0 \oplus b_0)
\end{aligned}
$$

$s = p_3 p_2 p_1 p_0$



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

Carry Skip Adder



The n-bit Carry Skip Adder consists of

a n-bit **carry-ripple-chain**,
a n-input **AND-gate** and
one **multiplexer**.

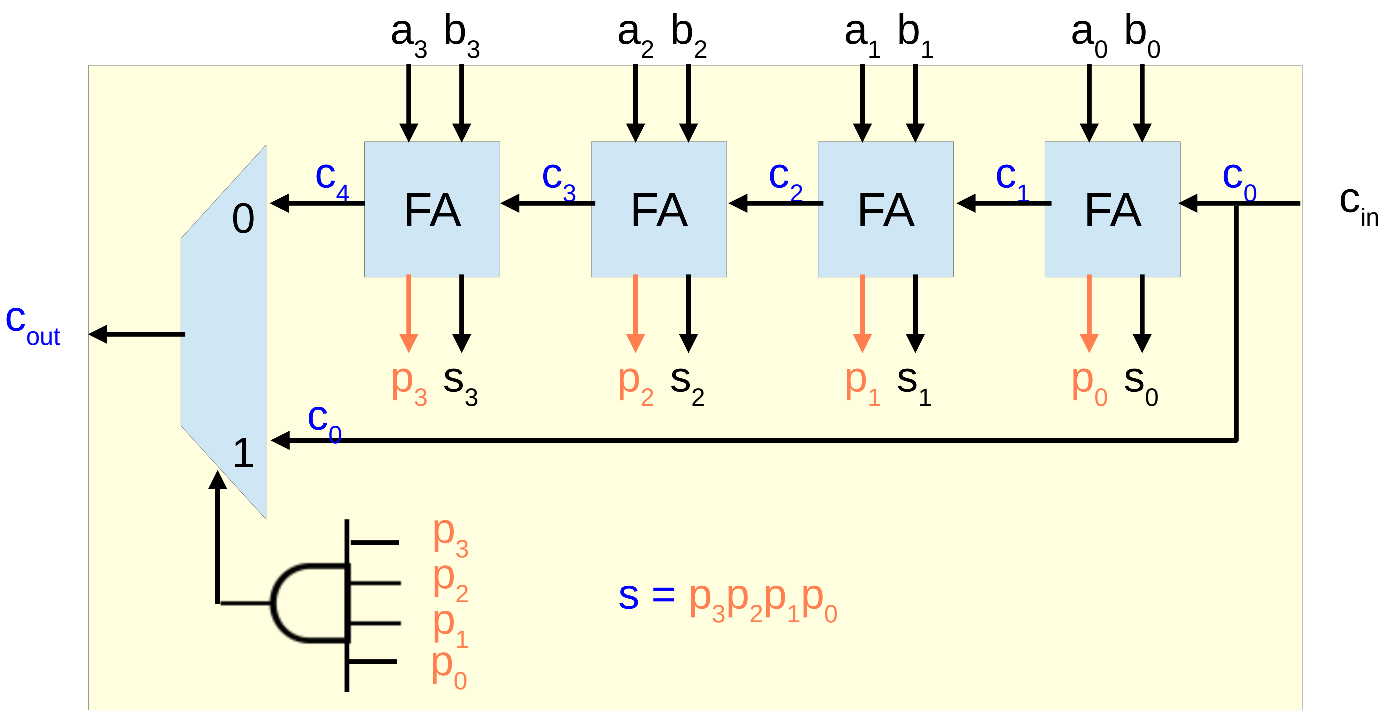
a multiplexer switches
either the last carry-bit c_n or the carry-in c_0
to the carry-out signal c_{out}

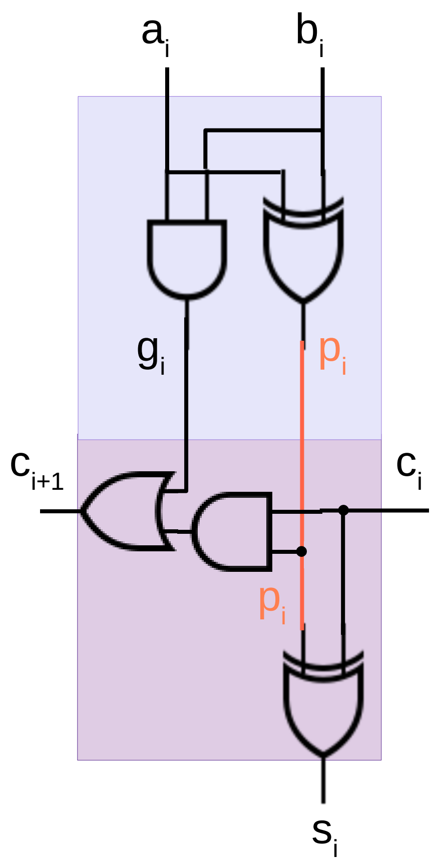
$s = p_3 \wedge p_2 \wedge p_1 \wedge p_0 = p_{[3:0]}$

when $s = 1$, $c_{out} \leftarrow c_0$
otherwise, internally generated carries
can be propagated to $c_{out} \leftarrow c_4$

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

Carry Skip Adder





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

Carry Skip Adder

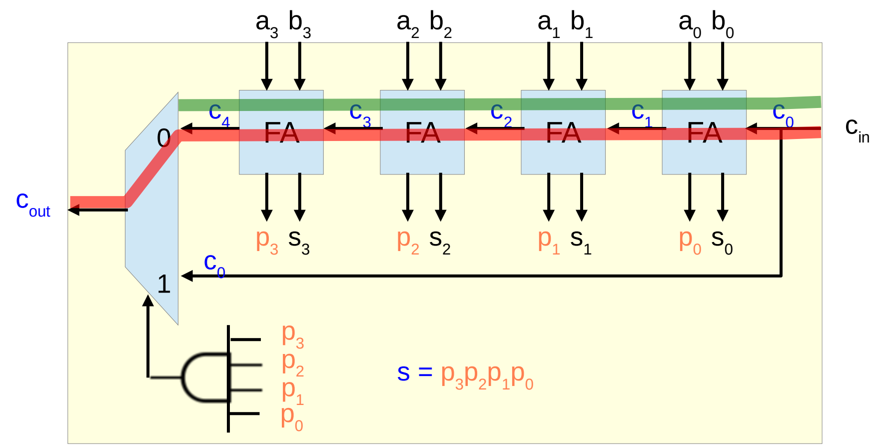
The critical path of a Carry Skip Adder begins at the first full adder, passes through all adders and ends at the sum bit s_{n-1}

Since a single *n-bit* Carry Skip Adder has no real speed benefit compared to a *n-bit* Ripple Carry Adder

$T_{CSA}(n) = T_{RCA}(n)$

$T_{RCA}(n)$

$T_{CSA}(n)$



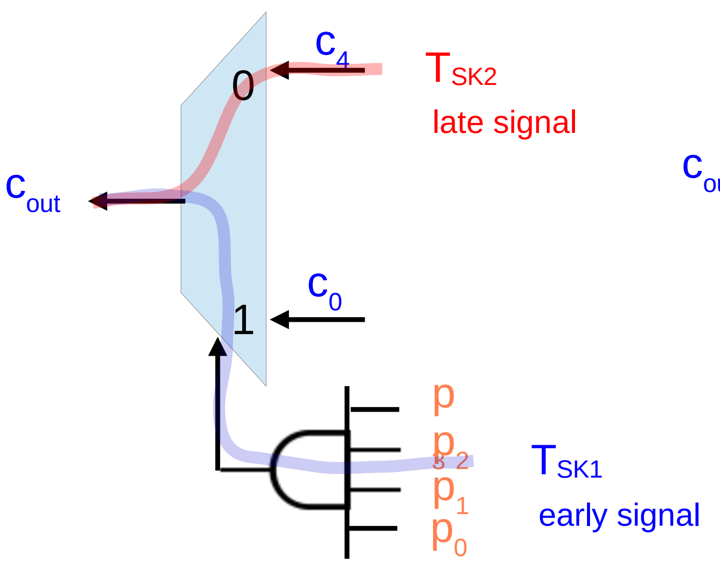
https://en.wikipedia.org/wiki/Carry-skip_adder

Carry Skip Adder

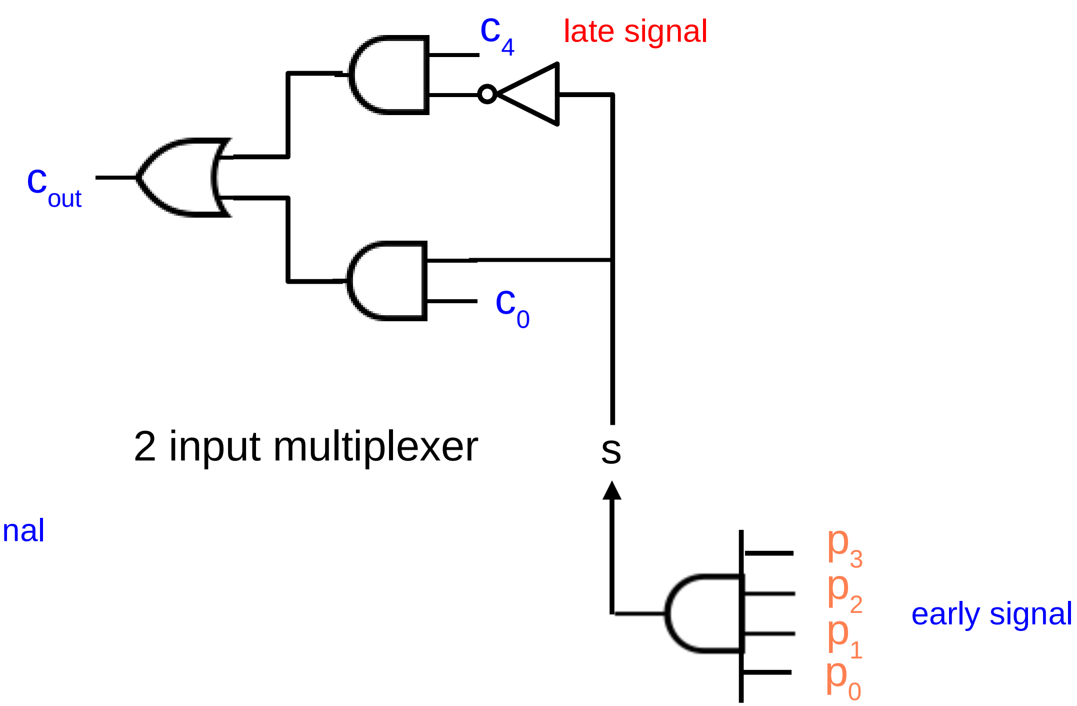
the <u>skip logic</u> consists of a k-input AND gate and one MUX

$T_{SK1} = T_{AND}(k) + T_{MUX}$ delay path through the AND gate

$T_{SK2} = T_{MUX}$ delay path from the ripple carry ... the critical path



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



2 input multiplexer

Carry Skip Adder

As the propagate signals
are computed in parallel and
are early available,

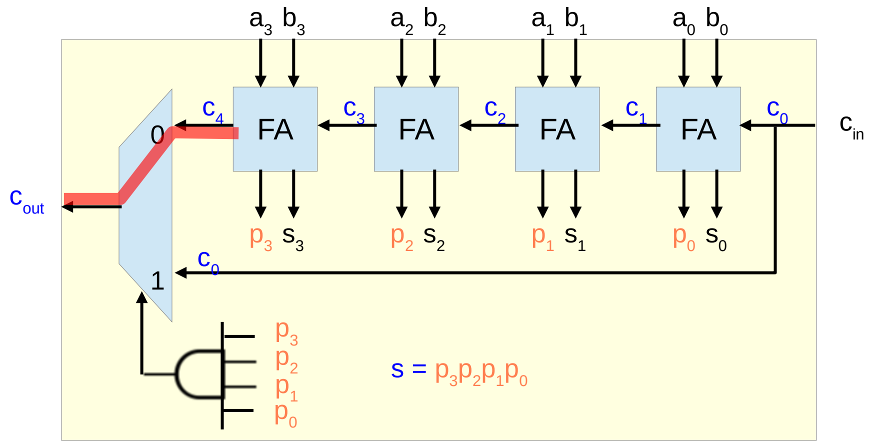
$p_i = a_i \oplus b_i$

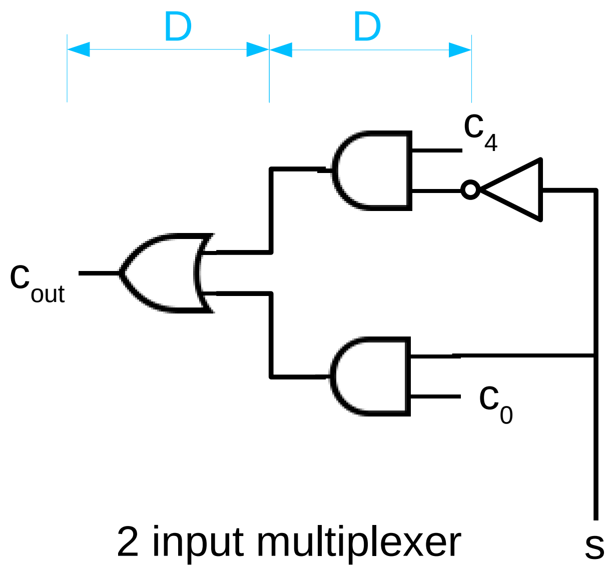
The critical path in a Carry Skip Adder consists of ripple carry path and mux path for ripple carry (T_{SK2})

the critical path for the skip logic in a Carry Skip Adder
consists of the delay imposed by the multiplexer (conditional skip)

T_{CSK} skip logic delay in the critical path

$T_{CSK} = T_{SK2} = T_{MUX} = 2D$





2 input multiplexer

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

Block Carry Skip Adder

Block carry skip adders are composed of a number of carry skip adders

There are two types of block carry skip adders

The two operands A = (a_{n-1}, a_{n-2}, … a_1, a_0) and B = (b_{n-1}, b_{n-2}, … b_1, b_0) are split in k blocks of (m_k, m_{k-1}, … m_2, m_1) bits

- Why are block carry skip adders used
- Should the block size be constant or variable?
- Fixed block size vs. variable block size

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Fixed-size Block Carry Skip Adder

Fixed-size block CSA (FCSA)

Carry Skip Adders are chained to reduce the overall critical path,
(Block Carry Skip Adders)

Fixed size block Carry Skip Adders (FCSA) split the *n* bits of the input bits
Into blocks of *k* bits each, resulting in R = *n* / *k* blocks.

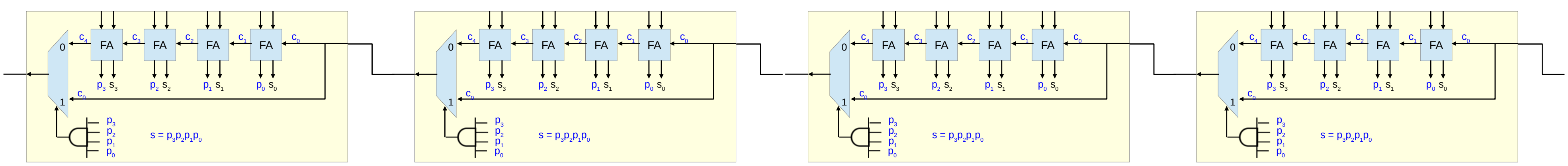
number of blocks

bits per block
block size

$$n = R \cdot k$$

R blocks

k bits	k bits	k bits	k bits
[(R)k-1 … (R-1)k] = [N-1 … N-k-1]	[(r+1)k-1 … (r)k] $r \in [R-1, \cdots, 0]$	[(2)k-1 … (1)k] = [2k-1 … k]	[(1)k-1 … (0)k] = [k-1 … 0]



Www.cs.tufts.edu

Carry Skip Adder

the critical path

the longest carry path must be

- <u>generate</u> in the <u>first</u> block
- <u>terminated</u> in the <u>last</u> block
- <u>propagated</u> in the blocks between the first and the last

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

R groups

k bits

Fixed-size block CSA (FCSA)

$$n = R \cdot k$$

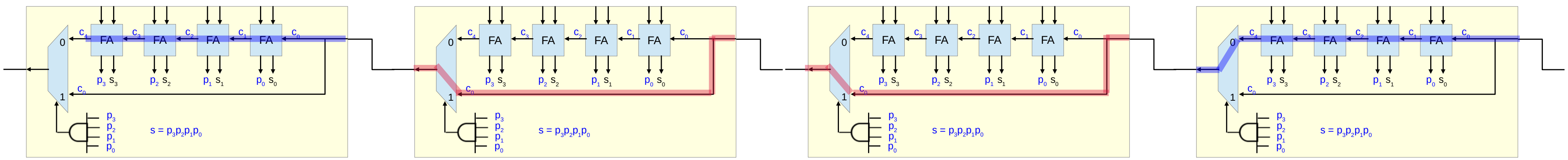
the last block

carry terminated in the last FA

(R-2) blocks

the first block

carry generated in the first FA



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

Carry Skip Adder

The critical path consists of

- the ripple path and the skip element of the first block
  $T_{CRA}(k)$ + T_{CSK} $T_{CRA[0:cout]}(k)$
- the skip paths that are enclosed between the first and the last block
  $(R-2)T_{CSK}$
- finally the ripple path of the last block
  $T_{CRA}(k)$

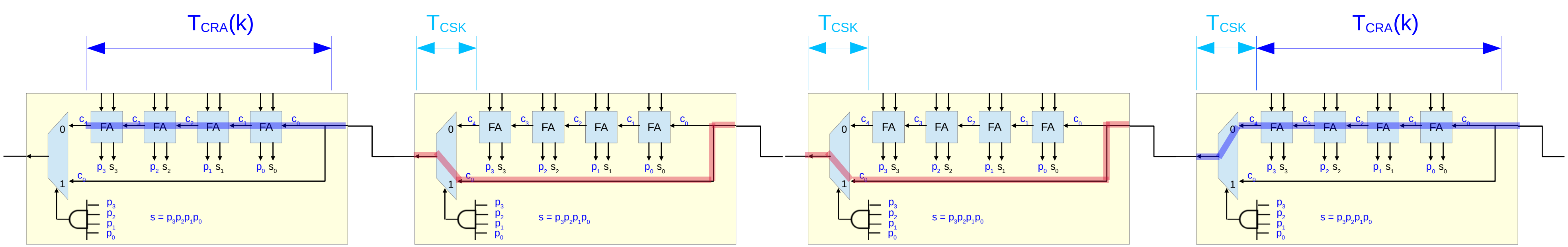
Fixed-size block CSA (FCSA)

R groups

k bits

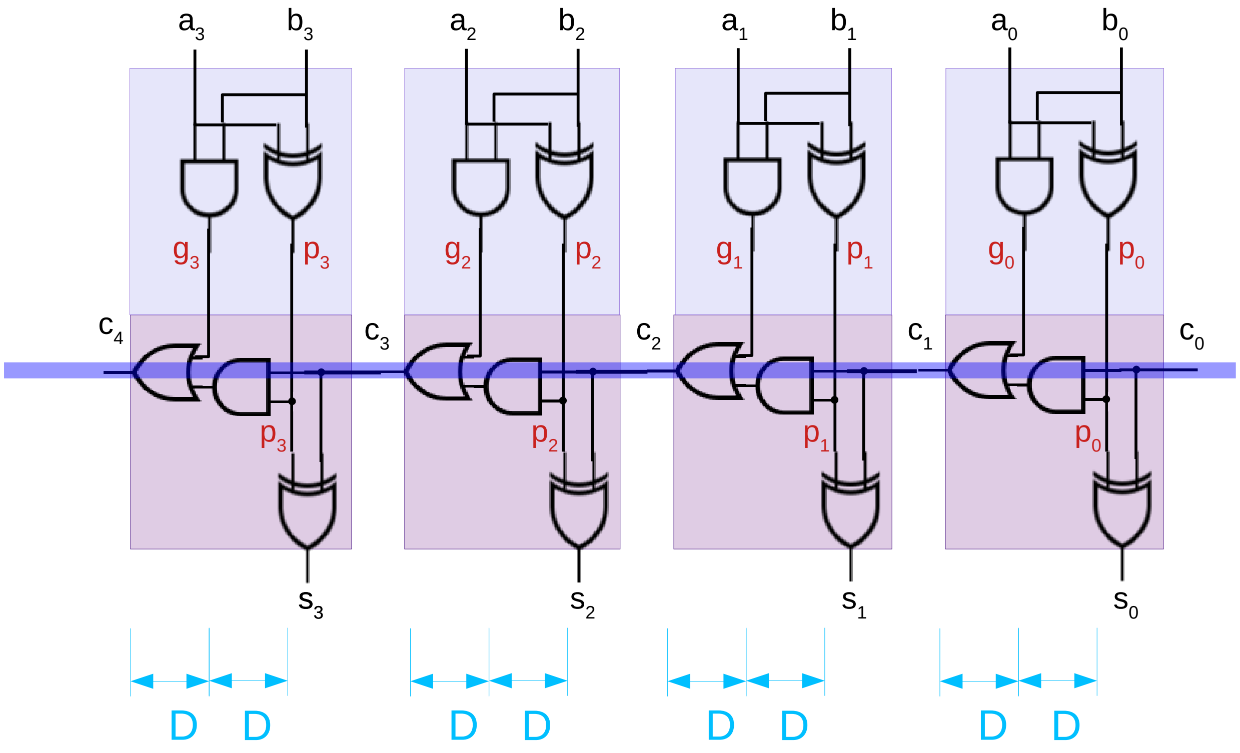
$$n = R \cdot k$$

the last block (R-2) blocks the first block



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

4-bit Full Adder with P and G



k bits

$T_{CRA}(k) = 2D \cdot k$

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

Critical Path Delay

Fixed-size block CSA (FCSA)

The critical path consists of

- the <u>ripple path</u> and the <u>skip element</u> of the first block $T_{CRA}(k) + T_{CSK}$
- the <u>skip paths</u> that are enclosed between the <u>first</u> and the <u>last</u> block $(R-2)T_{CSK}$
- finally the <u>ripple path</u> of the last block $T_{CRA}(k)$

$$
\begin{aligned}
T_{FCSA}(n) &= T_{CRA}(k) + T_{CSK} + (R-2)T_{CSK} + T_{CRA}(k) \\
&= k\,2D + 2D + (R-2)2D + k2D \\
&= k2D + 2D + (R-1)2D - 2D + k2D \\
&= k2D + (R-1)2D + k2D \\
&= 2k2D + (R-1)2D \\
&= (2k+R)2D \\
&=(2k+n/k)2D
\end{aligned}
$$

$$
\begin{aligned}
&= k\,2D + 3D + (R-1)2D + (k+2)2D \\
&= 3D + k2D + R2D - 2D + k2D + 4D \\
&= (2k+R)2D + 5D
\end{aligned}
$$

R groups

k bits

$$n = R \cdot k$$

Optimal block size k

$T_{FCSA}(n) = T_{CRA}(k) + T_{CSK} + (R-2)T_{CSK} + T_{CRA}(k)$
$= (2k+R)2D$
$= (2k+n/k)2D$

$$\left(2k + \frac{n}{k}\right)2D$$

The optimal block size k for a given adder width n

$dT_{FCSA}(n) / dk = 0$

$$\frac{dT_{FCSA}(n)}{dk} = 0$$

$(2-n(1/k^2)) = 0$
$2 = n/k^2$
$k^2 = n / 2$

$$k^2 = \frac{n}{2}$$

$$k_{opt} = \sqrt{\frac{n}{2}}$$

$k = \sqrt{n/2}$

$5.6 = \sqrt{64/2}$ n = 64bits → k = 6
$4 = \sqrt{32/2}$ n = 32bits → k = 4

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

Fixed-size block CSA (FCSA)

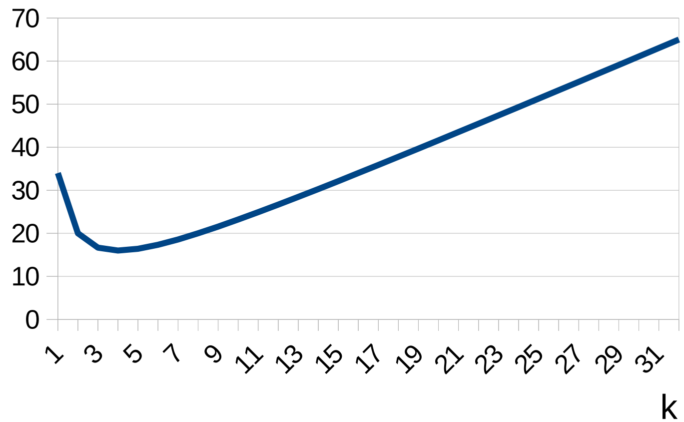
R groups
k bits
n = R·k

Examples of Optimal Block Sizes

$$T_{FCSA,opt}(n) = \left(2k + \frac{n}{k}\right)2D$$

$T_{FCSA}(32) = (2k+32/k)2D$

(2k+32/k)



$4 = \sqrt{32/2}$

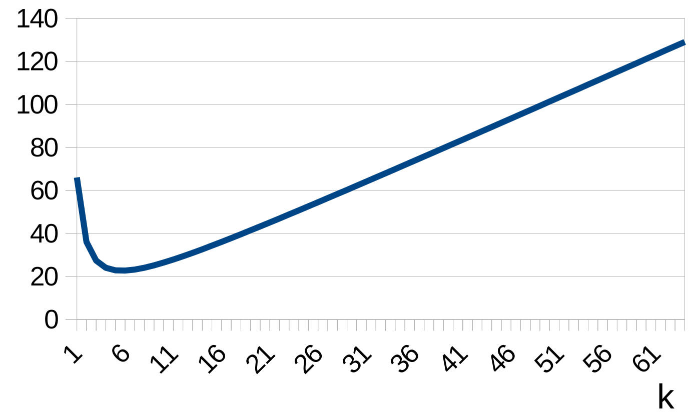
$$k_{opt} = \sqrt{\frac{n}{2}}$$

n = 32bits
→ k = 4

$$T_{FCSA,opt}(n) = \left(2k + \frac{n}{k}\right)2D$$

$T_{FCSA}(64) = (2k+64/k)2D$

(2k+64/k)



$5.6 = \sqrt{64/2}$

$$k_{opt} = \sqrt{\frac{n}{2}}$$

n = 64bits
→ k = 6

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

Fixed-size block CSA (FCSA)

R groups
k bits
n = R·k

Asymptotic Analysis

Fixed-size block CSA (FCSA)

R groups

k bits

n = R·k

$T_{FCSA}(n) = (2k+n/k)2D$

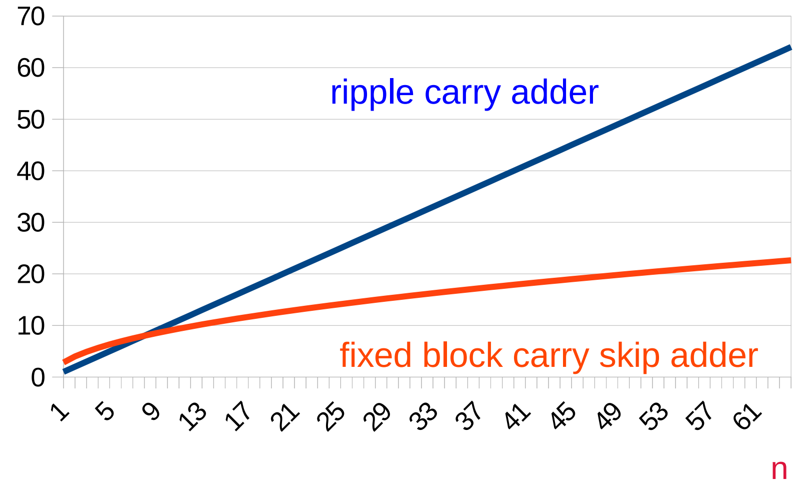
The optimal block size k for a given adder width n
$k = \sqrt{n/2}$

$$
\begin{aligned}
T_{FCSA,\,opt}(n) &= (2\sqrt{n/2} + n/\sqrt{n/2})2D \\
&= (\sqrt{2n} + \sqrt{n^2 / (n/2)}\,)\ 2D \\
&= (\sqrt{2n} + \sqrt{2n})2D \\
&= (2\sqrt{2n})2D
\end{aligned}
$$

$$
\begin{aligned}
T_{FCSA,opt}(n) &= \left(2\sqrt{n/2} + \frac{n}{\sqrt{n/2}}\right)2D \\
&= \left(2\sqrt{2n}\right)2D \qquad \textit{when} \quad k_{opt} = \sqrt{\frac{n}{2}}
\end{aligned}
$$

$T_{CRA}(n) = 2D \cdot n$

$T_{FCSA,opt}(n) = 2D \cdot (2sqrt(2n))$

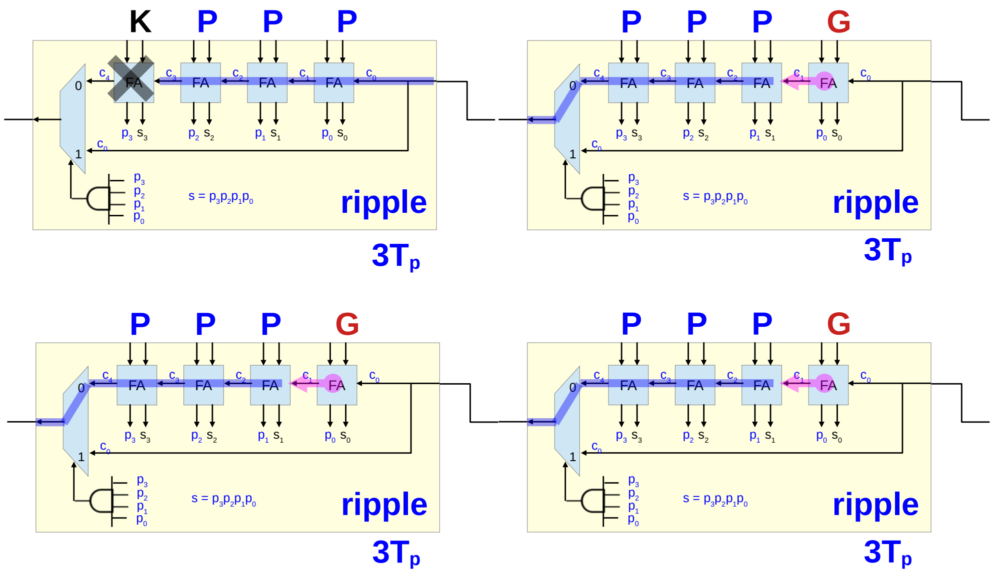


https://en.wikipedia.org/wiki/Carry-skip_adder

Carry Skip Adder

If an arbitrary block generated a carry by itself,
the carry will always propagate to the next block

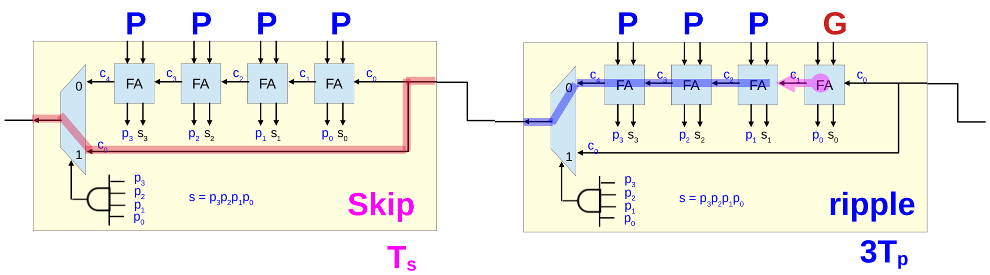
however, if the second block generates a carry itself,
or kill the carry, then that is the end of the critical path



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

Carry Skip Adder

If the second block propagates the carry,
then we see the advantage of the CSA architecture



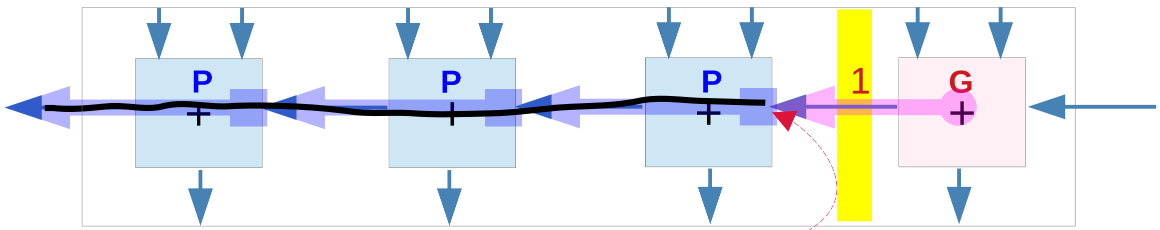
https:://electronics.stackexchange.com/questions/21251/critical-path-for-carry-skip-adder

Critical Carry Path (1)

$T_s < 3T_p$

For longest carry path, if any block <u>generates</u> a carry,
that carry will <u>propagate</u> through the remaining 3 FA's of that block

1



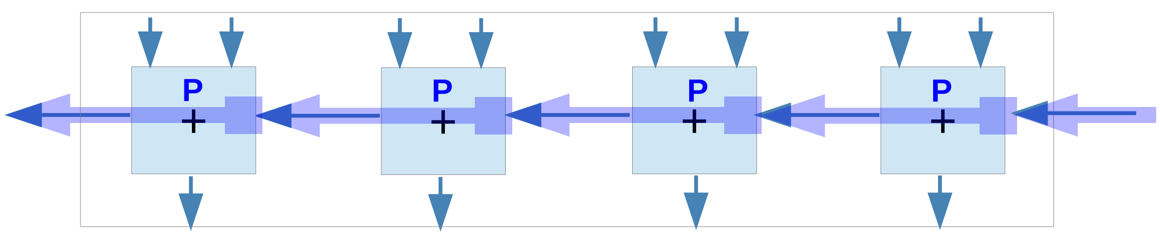
Least Significant Block

C_{in} = 1 or 0

carry delay path starts from this FA

and then through the carry skip gates to the <u>final</u> block,

1



Middle Blocks

1

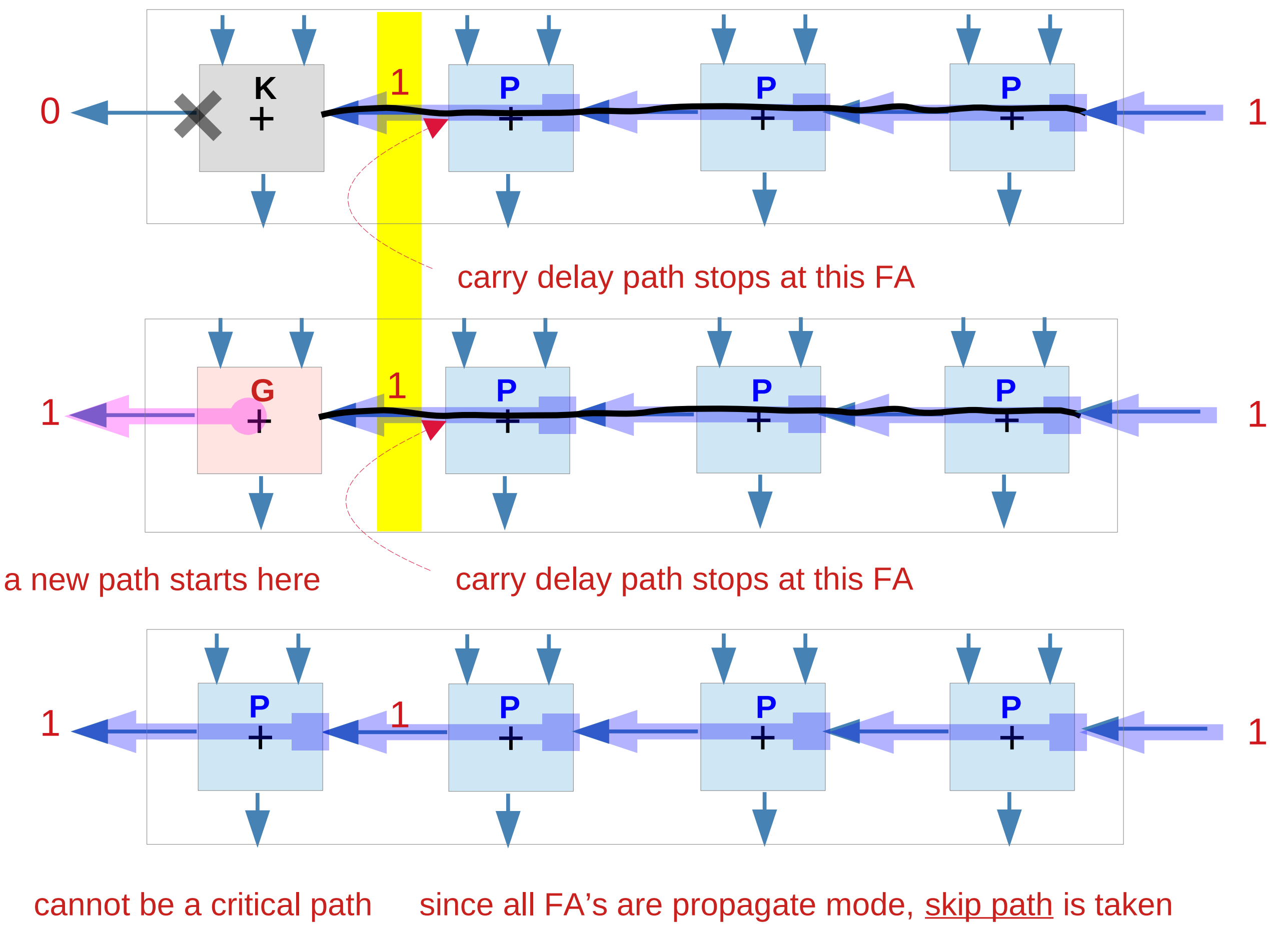
since all FA's are in the propagate mode, skip path is taken → no ripple delay

More Fast Adders, Ivor Page, University of Texas at Dallas

Critical Carry Path (2)

At the <u>final</u> block, it may have to <u>propagate</u> through 3 FA's to reach the most significant adder

Cases for the Most Significant Block



the propagated carry must be terminated at the most significant adder

the new carry can be generated at the most significant adder

Otherwise, the whole block is to be skipped

smaller delay than ripple delay

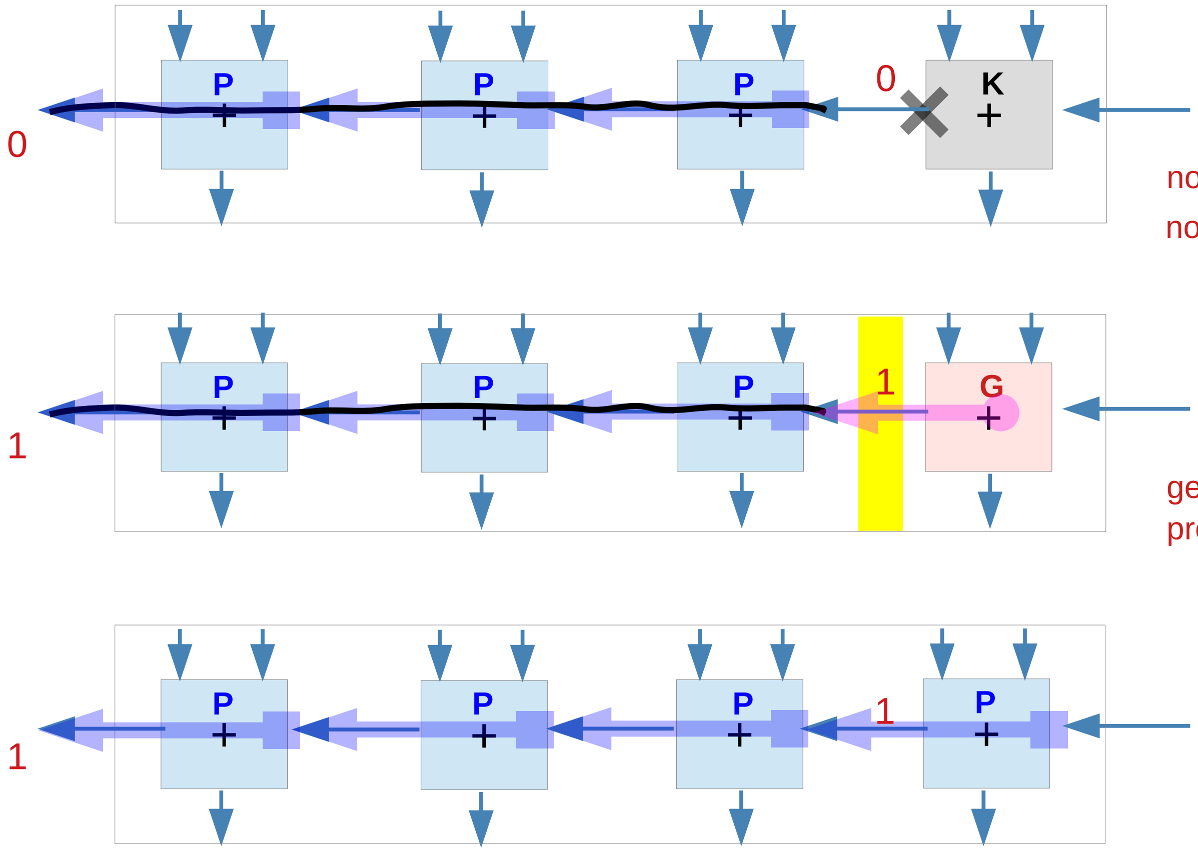
$T_s < 3T_p$

More Fast Adders, Ivor Page, University of Texas at Dallas

Critical Carry Path (3)

$T_s < 3T_p$

Cases for the Least Significant Block



no carry out

no carry propagation

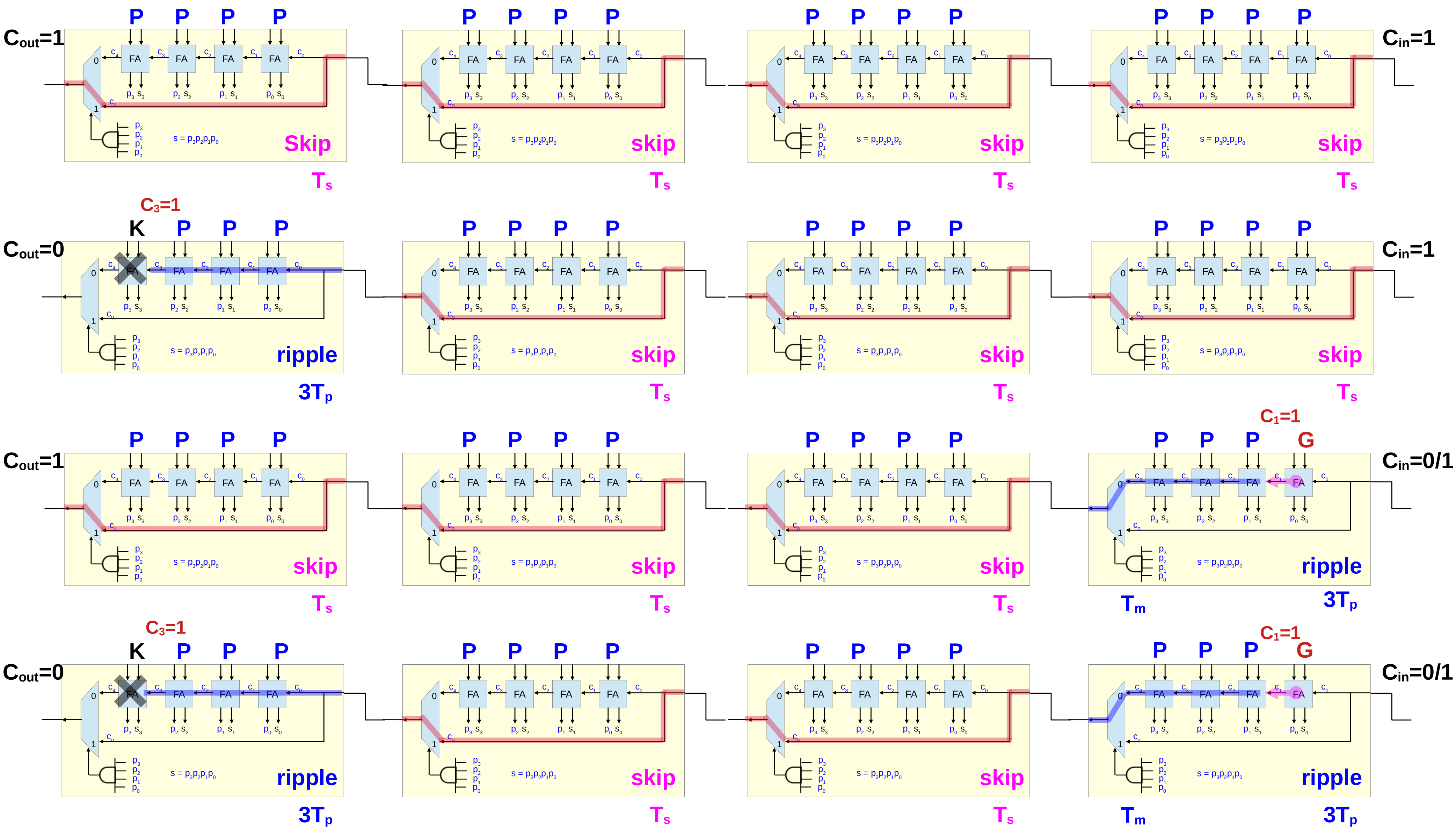
generated carry can be propagated to other blocks

$T_s < 3T_p$

cannot be a critical path since all FA's are propagate mode, skip path is taken → no ripple delay

More Fast Adders, Ivor Page, University of Texas at Dallas

Critical Carry Path (4)



Carry Skip Adder

The longest delay path from c_1 to c_{n-1}

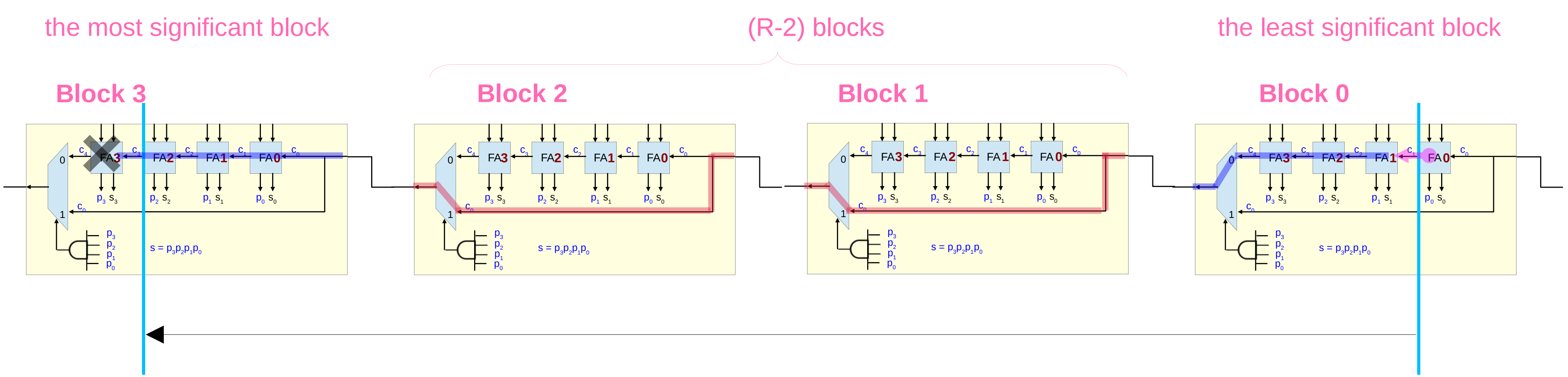
begins with a carry generated in FA0 in the least significant block 0,
propagates through FA3 in block 0,
then through the skip element (MUX can be replaced with OR gate),
then through carry skip units of (R-2) blocks,
and then through fa0, fa1, fa2 in the most significant block (R-1),
to the c_{n-1} signal

Fixed-size block CSA (FCSA)

R groups

k bits

$$n = R \cdot k$$



More Fast Adders, Ivor Page, University of Texas at Dallas

Carry Skip Adder

The longest delay path from c_1 to c_{n-1}

$(k-1)T_p + T_m + (n/k-2)T_s + (k-1)\ T_p$

T_p is the time to propagate a carry through one stage of the full adder (from c_i to c_{i+1})

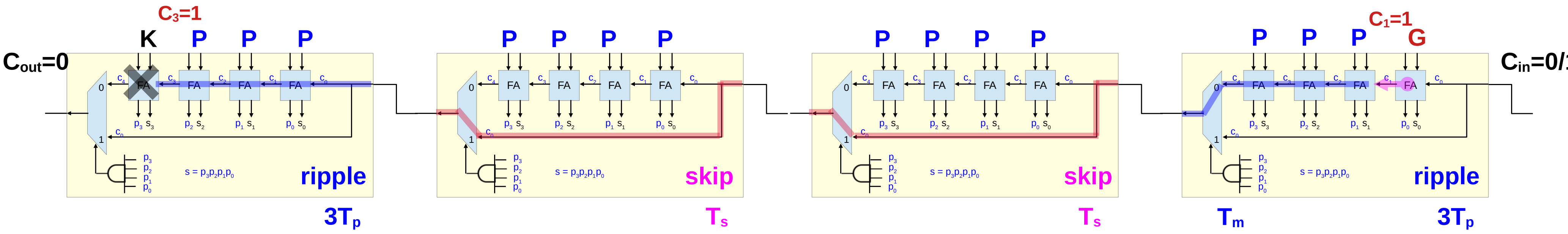
T_s is the delay through one carry-skip stage

Fixed-size block CSA (FCSA)

R groups

k bits

$n = R \cdot k$



More Fast Adders, Ivor Page, University of Texas at Dallas

Carry Skip Adder

The longest delay path from c_1 to c_{n-1}

$(k-1)T_p + T_m + (n/k-2)T_s + (k-1)\ T_p$

Carry Skip Adder is faster than RCA at the expense
of a few relatively simple modifications.

The delay is still linearly dependent on the size of the adder N,
however this linear dependence is reduced by a factor of *1/k*

$$T_{fixed-skip-add} = (b-1)T_p + D + (k/b-2)T_s + (b-1)T_p$$

The original formula in the literature

More Fast Adders, Ivor Page, University of Texas at Dallas

Fixed-size block CSA (FCSA)

R groups

k bits

n = R·k

1. k ← n : total number of bits
2. b ← k : block size in bits

Carry Skip Adder

T_p is the time to propagate a carry
through one stage of the adder (from c_i to c_{i+1}), and

T_s is the delay through one carry-skip stage

Recall that T_p = 2D in the standard ripple-carry adder
based on two half-adders.

The delay T_s = 2D since there is an AND gate and an OR gate
In series in the carry-skip unit.

$(k-1)T_p + T_m + (n/k-2)T_s + (k-1)\ T_p$
$= (k-1)2D + D + (n/k-2)2D + (k-1)\ 2D$
$= 2kD -2D + D + 2Dn/k -4D +2kD -2D$
$= 4kD + 2Dn/k -7D$

$T_{fixed-skip-add} = 4Dk + 2nD/k - 7D$

More Fast Adders, Ivor Page, University of Texas at Dallas

Fixed-size block CSA (FCSA)

R groups

k bits

$n = R \cdot k$

1. k ← n : total number of bits
2. b ← k : block size in bits

Carry Skip Adder

Fixed-size block CSA (FCSA)

R groups

k bits

$n = R \cdot k$

The optimum block size, b^{opt} , is found
by differentiating the right-hand side
with respect to b and equating the result to zero.

$T_{fixed-skip-add} = 4Dk + 2nD/k - 7D$

$d\,T_{fixed-skip-add} / d\,k = d\,(4Dk + 2nD/k - 7D) / d\,k$
$= 4D - 2nD/k^2 = 0$

$4D = 2nD/k^2$
$k^2 = n/2$

$k^{opt} = \sqrt{n/2}$

More Fast Adders, Ivor Page, University of Texas at Dallas

1. k ← n : total number of bits
2. b ← k : block size in bits

Carry Skip Adder

Fixed-size block CSA (FCSA)

R groups

k bits

$n = R \cdot k$

The corresponding adder delay is:

$$
\begin{aligned}
T^{opt}_{fixed-skip-add} &= 4Dk^{opt} + 2nD/k^{opt} - 7D \\
&= 4D\sqrt{n/2} + 2nD/\sqrt{n/2} - 7D \\
&= (4D\sqrt{n})/\sqrt{2} + 2\sqrt{2}nD/\sqrt{n}) - 7D \\
&= (4/\sqrt{2}D\sqrt{n} + 2\sqrt{2}D\sqrt{n}) - 7D \\
&= (4/\sqrt{2} + 2\sqrt{2})\, D\sqrt{n} - 7D \\
&= 5.66\, D\, \sqrt{n} - 7D
\end{aligned}
$$

$k^{opt} = \sqrt{n/2}$

For example, in a n = 32 bit adder,

$k^{opt} = \sqrt{32/2} = \sqrt{16} = 4$ and

the delay is approximately 25D.

$5.66\sqrt{32} - 7 = 25.0$

Compare this value with the delay of a ripple-carry system, 64D.

More Fast Adders, Ivor Page, University of Texas at Dallas

1. k ← n : total number of bits
2. b ← k : block size in bits

Carry Skip Adder

In a n = 64 bit adder, $k^{opt} = \sqrt{n/2} = \sqrt{32} = 5.657$.

If we use k = 4, the delay is 41D.

$4Dk + 2nD/k - 7D = 16D + 2*64/4\ D - 7D = (16+32-7)D = 41D$

If k = 8 the delay is again 41D.

$4Dk + 2nD/k - 7D = 32D + 2*64/8\ D - 7D = (32+16-7)D = 41D$

An in-between solution is possible with k = 6.

$4Dk + 2nD/k - 7D = 24D + 2*60/6\ D - 7D = (24+20-7)D = 37D$

Then there are 10 blocks of 6 and 1 block of 4

64 = 6*10 + 4

(at the most significant end).

The corresponding delay is 35D. → (33D)

The 64 bit ripple-carry adder has delay 128D.

More Fast Adders, Ivor Page, University of Texas at Dallas

Fixed-size block CSA (FCSA)

R groups

k bits

n = R·k

$(k-1)T_p + T_m + (n/k-2)T_s + (k-1)\ T_p$

$= (6-1)2D + D + (10-2)2D + (4-1)\ 2D$

$= (10 + 1 + 16 + 6)D$

$= 33D$

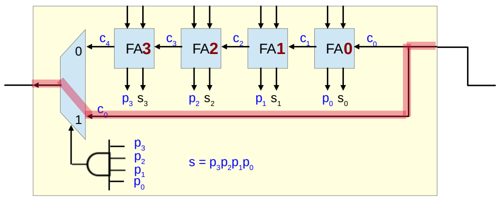
(6-1)2D → (4-1)2D : -4D

37D – 4D = 33D

1. k ← n : total number of bits

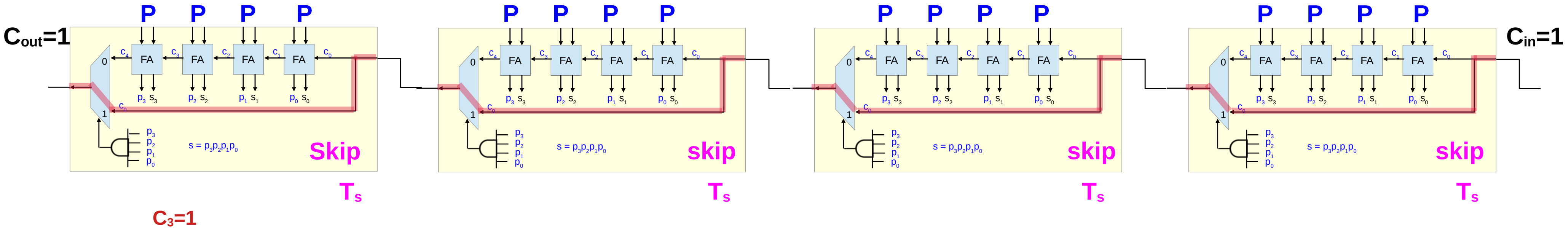
2. b ← k : block size in bits

Carry Skip Adder



A carry signal centering a certain block
can be propagated past the block
<u>without</u> <u>waiting</u> for the signal to propagate
through the 4 individual stages of the block

If all n/4 blocks propagate, a carry entering the least significant stage
will pass to the most significant carry-out
in time n/4 times the delay through the carry-skip unit



More Fast Adders, Ivor Page, University of Texas at Dallas

Variable size Block Carry Skip Adder (1)

The performance can be improved, ie.
all carries propagated quickly
by varying the block sizes

Accordingly the initial blocks of the adder are
made smaller so as to quickly detect carry generates
that must be propagated the furthers,

the middle blocks are made larger
because they are not the problem case,

and then the most significant blocks are again
made smaller so that the late arriving carry inputs
can be processed quickly

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

Variable size Block Carry Skip Adder (2-1)

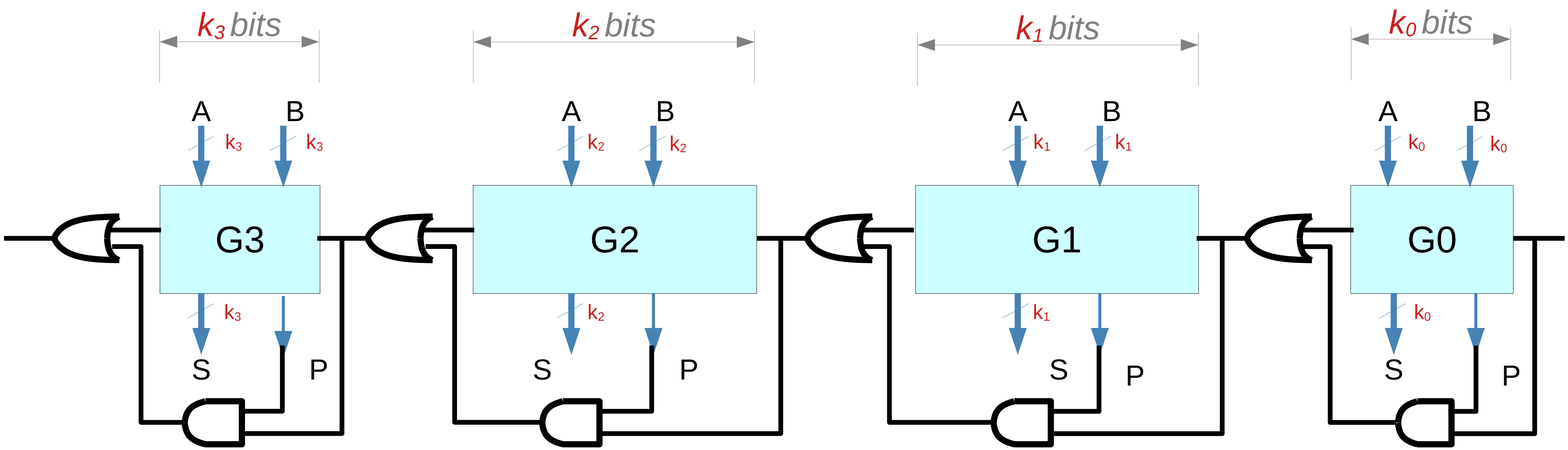
In the next development there are R carry-skip blocks with sizes $k_{R-1}, \cdots, k_1, k_0$ $(n = k_{R-1} + \cdots + k_1 + k_0)$ going from left to right.

Fixed-size block CSA (FCSA)

R groups

k bits

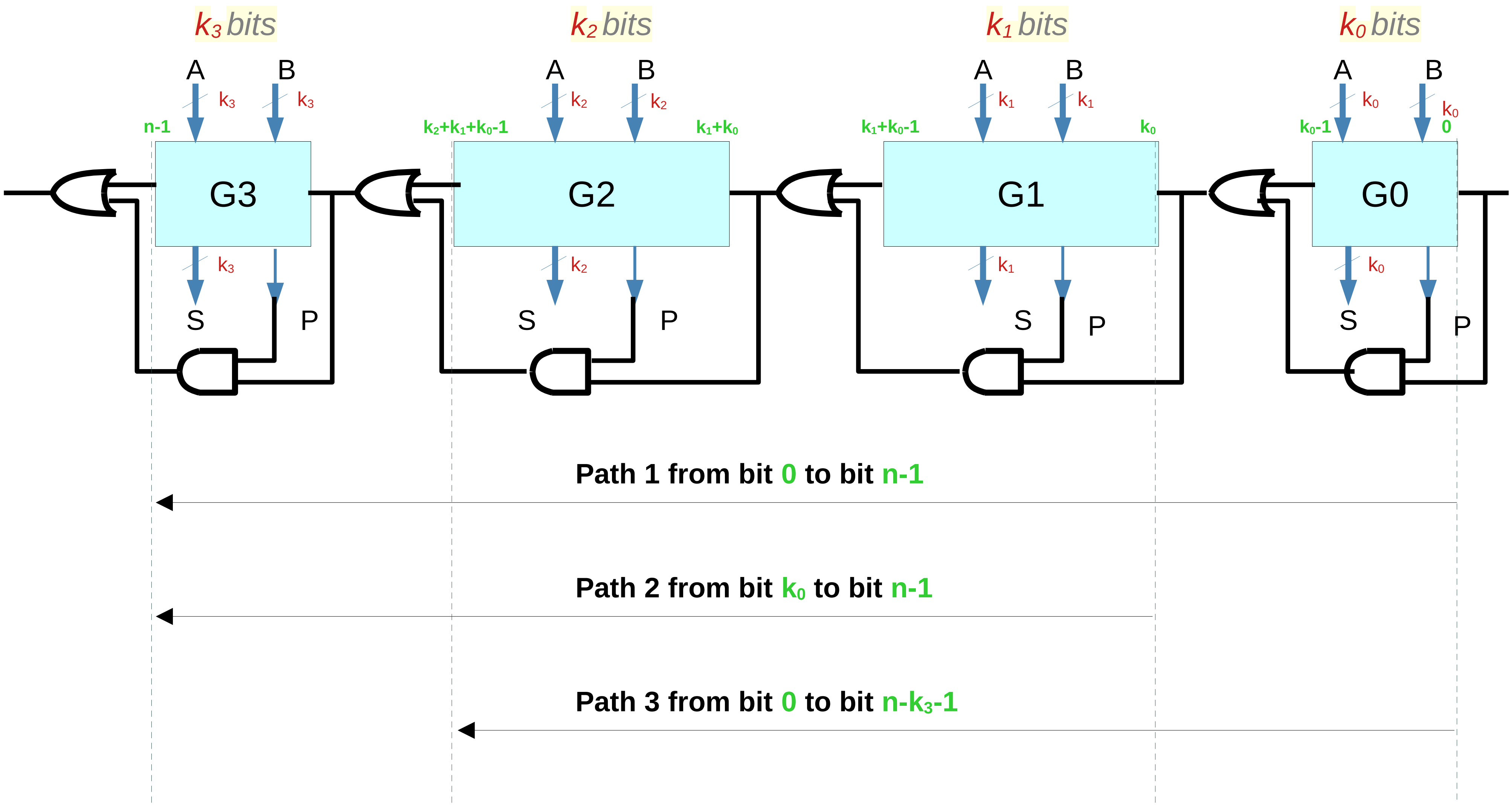
$n = R \cdot k$



More Fast Adders, Ivor Page, University of Texas at Dallas

1. k ← n : total number of bits
2. b ← k : block size in bits

Variable size Block Carry Skip Adder (2-1)



More Fast Adders, Ivor Page, University of Texas at Dallas

1. k ← n : total number of bits
2. b ← k : block size in bits

Variable size Block Carry Skip Adder (2-2)

Consider the equation for the worst case delay from stage 0 to stage n-1, corresponding to path 1

$T_{var-carry-skip} = T_{path\ 1}$

$= (k_{R-1} - 1)T_p + (R - 2)T_s + D + (k_0 - 1)T_p$

last block middle blocks OR first block

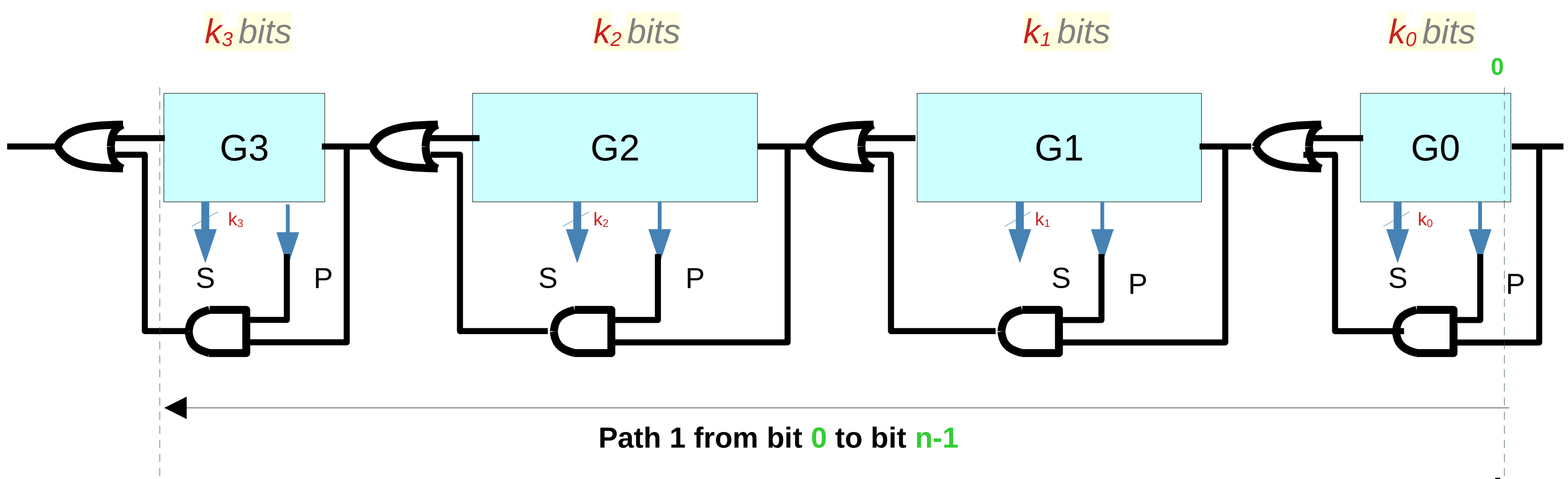
$n = k_{R-1} + \cdot \cdot \cdot + k_1 + k_0$

Fixed-size block CSA (FCSA)

R groups

k bits

$\mathbf{n = R \cdot k}$



More Fast Adders, Ivor Page, University of Texas at Dallas

1. k ← n : total number of bits
2. b ← k : block size in bits

Variable size Block Carry Skip Adder (2-2)

$T_{var-carry-skip} = T_{path\ 1}$

$= (k_{R-1} - 1)T_p + (R - 2)T_s + D + (k_0 - 1)T_p$

last block middle blocks OR first block

- a carry being generated by stage 0 in block G0 propagating through the $(k_0 - 1)$ remaining stages of block G0,
- then through the carry skip units of (R-2) blocks,
- then through $(k_{R-1} - 1)$ stages of the left-most (last) block.

Fixed-size block CSA (FCSA)

R groups

k bits

$$n = R \cdot k$$

More Fast Adders, Ivor Page, University of Texas at Dallas

1. k ← n : total number of bits
2. b ← k : block size in bits

Variable size Block Carry Skip Adder (3-1)

Consider a carry being generated at the stage k_0,
the right-most stage of block 1,

and following path 2 to the left-most stage n-1 of the adder.
It's delay would be:

$$T_{path\ 2} = (k_{R-1} - 1)T_p + (R - 3)T_s + D + (k_1 - 1)T_p$$

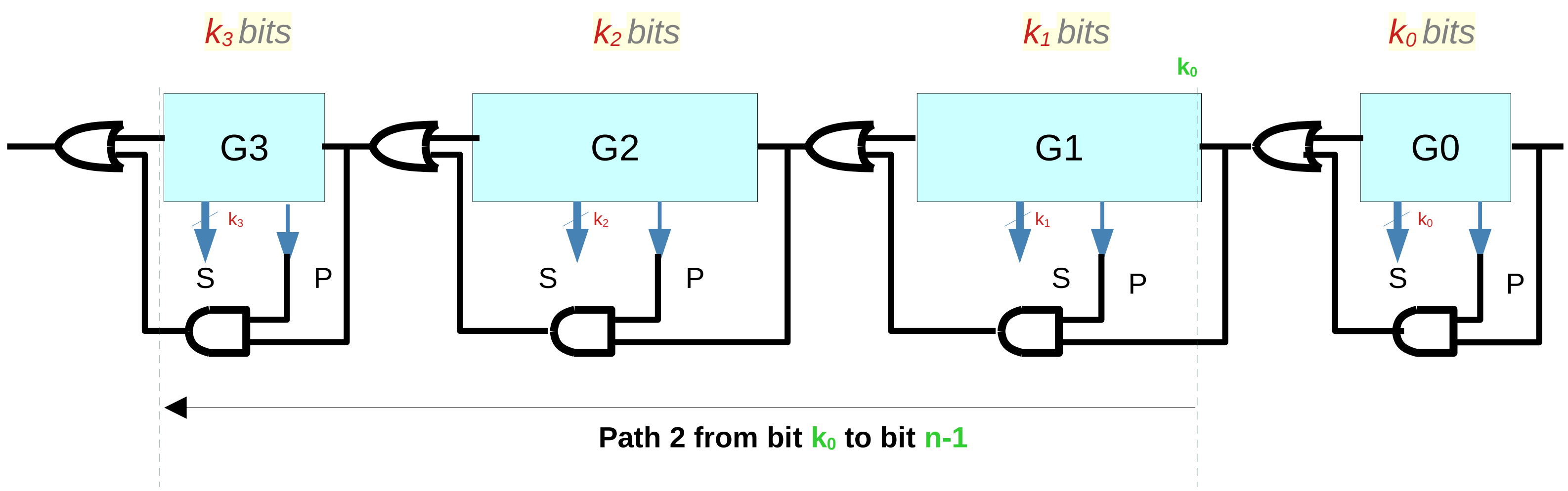
last block middle blocks OR first block

Fixed-size block CSA (FCSA)

R groups

k bits

$n = R \cdot k$



More Fast Adders, Ivor Page, University of Texas at Dallas

Variable size Block Carry Skip Adder (3-1)

Fixed-size block CSA (FCSA)

R groups
k bits
$\mathbf{n = R \cdot k}$

Path 1 has R blocks and R-2 middle blocks
Path 2 has R-1 blocks and R-3 middle blocks

$T_{path\ 1} = (k_{R-1} - 1)T_p + (R - 2)T_s + D + (k_0 - 1)T_p$
$T_{path\ 2} = (k_{R-1} - 1)T_p + (R - 3)T_s + D + (k_1 - 1)T_p$

$T_{path\ 1} - T_{path\ 2} = T_s + (k_0 - k_1)\ T_p \quad = 0$

If $T_s = T_p$ then block G1 with the size of k_1 bits
can be 1 bit larger than block G0 with the size of k_0 bits
without making this delay path worse than path 1.

$T_s + (k_0 - k_1)\ T_p \quad = (1 + (k_0 - k_1))T_p \quad = 0$

$k_1 = k_0 + 1$

More Fast Adders, Ivor Page, University of Texas at Dallas

1. k ← n : total number of bits
2. b ← k : block size in bits

Variable size Block Carry Skip Adder (3-2)

Similarly, if $k_2 = k_1 + 1$, the worst case delay from stage ($k_0 + k_1$) to stage ($n - 1$) will be no larger than the delay for path 1.

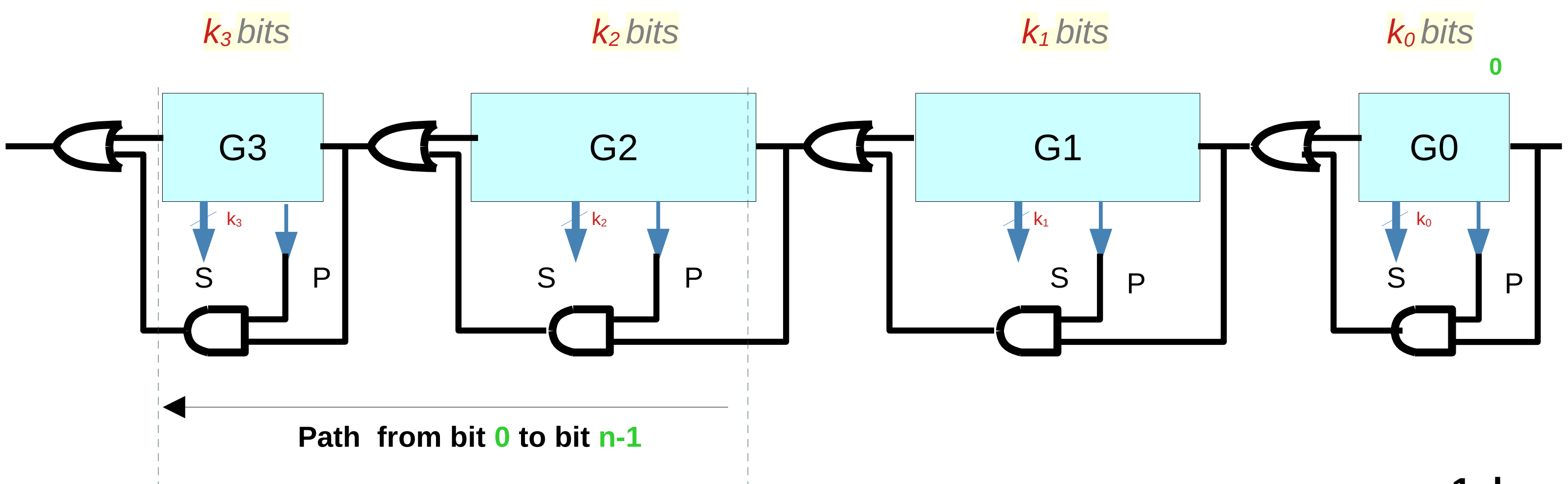
Blocks to the right of the center of the adder may therefore have sizes that form a simple incremental sequence.

Fixed-size block CSA (FCSA)

R groups

k bits

$n = R \cdot k$



More Fast Adders, Ivor Page, University of Texas at Dallas

1. k ← **n :** total number of bits
2. b ← **k :** block size in bits

Variable size Block Carry Skip Adder (3-3)

Similarly, if $k_2 = k_1 + 1$, the worst case delay
from stage ($k_0 + k_1$) to stage ($n - 1$)
will be no larger than the delay for path 1.

Blocks to the right of the center of the adder
may therefore have sizes
that form a simple incremental sequence.

Now consider a carry being generated in stage 0
and used (absorbed) in the left-most stage
of the penultimate block of the adder,
stage $k - b_{t-1} - 1$.

This corresponds to path 3 in the diagram.
Its delay is:
$T_{path\ 3} = (b_{t-2} - 1)T_p + (t - 3)T_s + D + (b_0 - 1)T_p$

Fixed-size block CSA (FCSA)

R groups

k bits

$n = R \cdot k$

1. k ← n : total number of bits
2. b ← k : block size in bits

More Fast Adders, Ivor Page, University of Texas at Dallas

Variable size Block Carry Skip Adder (4-1)

Compare this with the delay for path 1,
the longest carry-propagation path.

Again, if $T_p = T_s$, block size b_{t-2}
can be one larger than block size b_{t-1}
without making this delay path worse than path 1.

Blocks to the left of the center of the adder
may also have sizes that form a simple incremental sequence.

This analysis suggests an organ-pipe structure for the block sizes,
b, b + 1, · · ·, (b+t)/2− 1, (b+t)/2 − 1, · · ·, b + 1, b

Fixed-size block CSA (FCSA)

R groups

k bits

$n = R \cdot k$

More Fast Adders, Ivor Page, University of Texas at Dallas

1. **k ← n : total number of bits**
2. **b ← k : block size in bits**

Variable size Block Carry Skip Adder (4-1)

Fixed-size block CSA (FCSA)

R groups

k bits

$\mathbf{n = R \cdot k}$

We examine the effects of such a structure with an example.

Consider a 28 bit adder with carry-skip block sizes 2,3,4,5,5,4,3,2.

The following tables show the worst case delay paths
for a carry generated in stage 0 and absorbed in stage i,
and for a carry generated in stage j and absorbed in stage 27.

Delay from stage 0 to stage i

i =	4	8	13	18	22	25	27
	7D	11D	15D	17D	17D	17D	17D

Delay from stage j to stage 27

j =	0	2	5	9	14	19	23
	17D	17D	17D	17D	15D	11D	7D

More Fast Adders, Ivor Page, University of Texas at Dallas

1. k ← n : total number of bits
2. b ← k : block size in bits

Variable size Block Carry Skip Adder (5-1)

The worst case delays are from carries generated
in stage zero of the adder and
absorbed anywhere in the left-hand half,
and from carries generated anywhere
in the right-hand half and absorbed in stage 27.

These eight delays of 17D are made equal
by making the block sizes vary in the organ-pipe fashion
described above.

Fixed-size block CSA (FCSA)

R groups

k bits

$n = R \cdot k$

More Fast Adders, Ivor Page, University of Texas at Dallas

1. **k ← n : total number of bits**
2. **b ← k : block size in bits**

Variable size Block Carry Skip Adder (5-2)

Fixed-size block CSA (FCSA)

R groups

k bits

$n = R \cdot k$

The total number of bits in the t blocks is then:

$2[b + (b + 1) + \cdot \cdot \cdot + (b + t/2 - 1)] = t(b + t/4 - 1/2)$

which gives:
$b = k/t - t/4 + 1/2$

The worst-case delay through the adder
with variable block sizes is then:

$T_{var-skip-add} = (b - 1)T_p + (t - 2)T_s + D + (b - 1)T_p$
= last block middle blocks OR gate first block
$= 4bD + 2tD - 7D$

More Fast Adders, Ivor Page, University of Texas at Dallas

1. k ← n : total number of bits
2. b ← k : block size in bits

Variable size Block Carry Skip Adder (6-1)

The optimal number of blocks is calculated as follows:

$dT_{var-skip-add} / dt = -4kD/t^2 + D = 0$

$t^{opt} = 2\sqrt{k}$

opt

$T_{var-skip-add}^{opt} = 4D\sqrt{k} - 5D$

which is approximately $\sqrt{2}$ smaller than with fixed block size.

Fixed-size block CSA (FCSA)

R groups

k bits

n = R·k

More Fast Adders, Ivor Page, University of Texas at Dallas

1. k ← n : **total number of bits**
2. b ← k : **block size in bits**

Variable size Block Carry Skip Adder (6-2)

Example:
Continuing with our 32 bit adder example of the previous section,

$T^{opt} = 2\sqrt{32} = 11.3$

If we choose t = 10, then b = 32/t − t/4 + 1/2 = 1.2.

Say we choose b = 1.

Fixed-size block CSA (FCSA)

R groups

k bits

$\mathbf{n = R \cdot k}$

More Fast Adders, Ivor Page, University of Texas at Dallas

1. k ← n : total number of bits
2. b ← k : block size in bits

Variable size Block Carry Skip Adder (6-3)

The block sizes are then 1,2,3,4,5,5,4,3,2,1, which only covers 30 bits.
Its delay is 17D.

The adder with block sizes 1,1,2,3,4,5,5,4,3,2,1,1 has delay
$1 \times 2D + D + 10 \times 2D + 1 \times 2D = 25D$.

The adder with block sizes 2,2,3,4,5,5,4,3,2,2 has delay
$1 \times 2D + D + 8 \times 2D + 1 \times 2D = 21D$.

The adder with block sizes 1,4,5,6,6,5,4,1 also has delay 21D.

Fixed-size block CSA (FCSA)

R groups

k bits

$n = R \cdot k$

More Fast Adders, Ivor Page, University of Texas at Dallas

1. **k ← n :** total number of bits
2. **b ← k :** block size in bits

Variable size Block Carry Skip Adder (7-1)

Fixed-size block CSA (FCSA)

R groups

k bits

$n = R \cdot k$

Notice here that the worst case delay
corresponds to a delay path
from the right-most stage of the right block of 6
to the left-most stage of the left block of 6.

As we discovered in the analysis,
there is a balance between the largest (or smallest) block size
and the number of blocks.

Compare these results with 35D obtained with fixed block sizes
and 128D obtained with a ripple-carry adder.

Our analysis gives optimal delay of 17.6D,
but we were only able to achieve 21D.

More Fast Adders, Ivor Page, University of Texas at Dallas

1. **k ← n : total number of bits**
2. **b ← k : block size in bits**

Variable size Block Carry Skip Adder (7-2)

For a 64 bit adder, the best sequence of block sizes appears to be:
2,4,5,6,7,8,8,7,6,5,4,2

and it has delay 29D
which is close to the optimum of 27D
for this type of adder.

Compare with 35D for the fixed block size adder
and 128D for the ripple-carry adder.

Fixed-size block CSA (FCSA)

R groups

k bits

$$n = R \cdot k$$

More Fast Adders, Ivor Page, University of Texas at Dallas

1. k ← n : total number of bits
2. b ← k : block size in bits

Variable size Block Carry Skip Adder (7-3)

Further developments are possible with the carry-skip idea
if more than one level of skip units is employed.
We shall not study these developments.

The text has an example of a 30 bit adder
with two levels of carry-skip units and
a delay of 17D according to my calculations,
which is no better than the single layer scheme
for a 30 bit adder developed above.

Fixed-size block CSA (FCSA)

R groups

k bits

$$n = R \cdot k$$

More Fast Adders, Ivor Page, University of Texas at Dallas

1. k ← n : total number of bits
2. b ← k : block size in bits

References

[1] en.wikipedia.org
[2] Parhami, “Computer Arithmetic Algorithms and Hardware Designs”