Carry Skip Adder (5A)

-
-

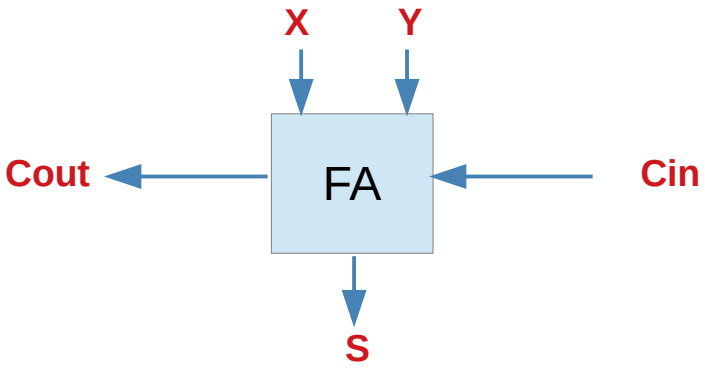
https://en.wikipedia.org/wiki/AND_gate
https://en.wikipedia.org/wiki/OR_gate
https://en.wikipedia.org/wiki/XOR_gate
https://en.wikipedia.org/wiki/NAND_gate

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

This document was produced by using OpenOffice and Octave.

Carry Kill, Propagate, Generate conditions (1)

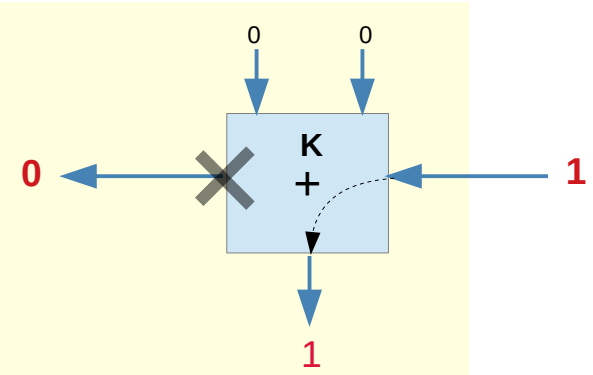
X	Y		
0	0	K	Kill (=$\overline{P}\overline{G}$)
0	1	P	Propagate
1	0	P	Propagate
1	1	G	Generate



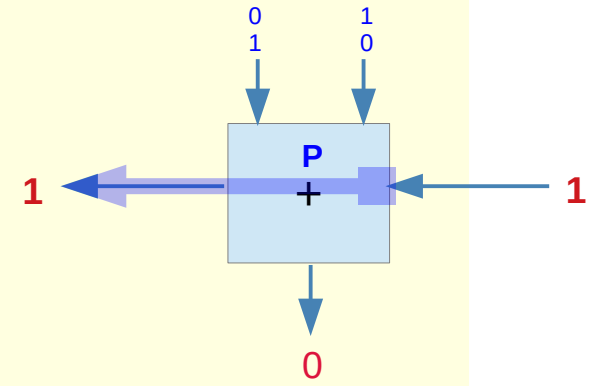
https::/electronics.stackexchange.com/questions/21251/critical-path-for-carry-skip-adder

Carry Kill, Propagate, Generate conditions (2)

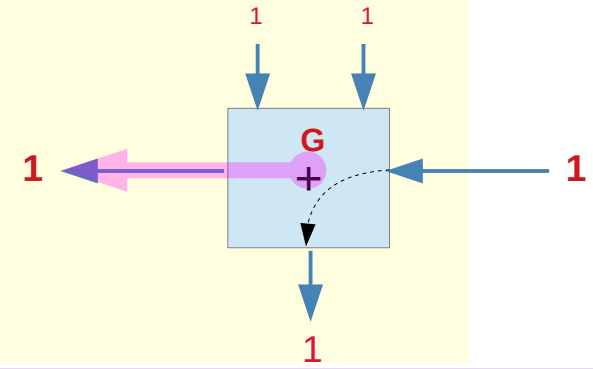
killed carry



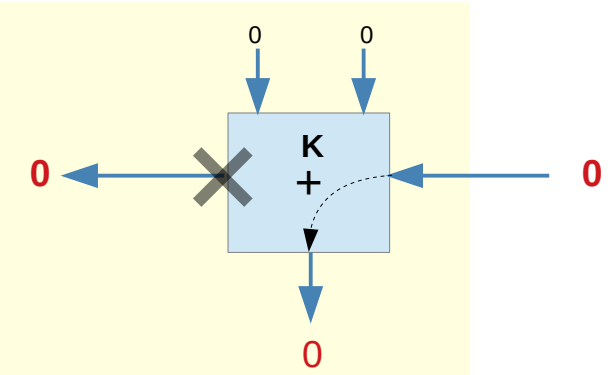
propagated carry



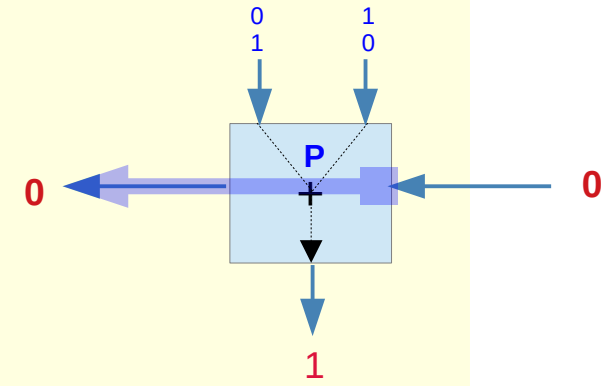
generated carry



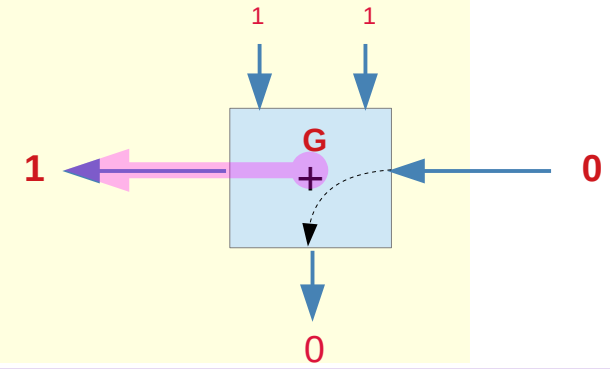
killed carry



propagated carry

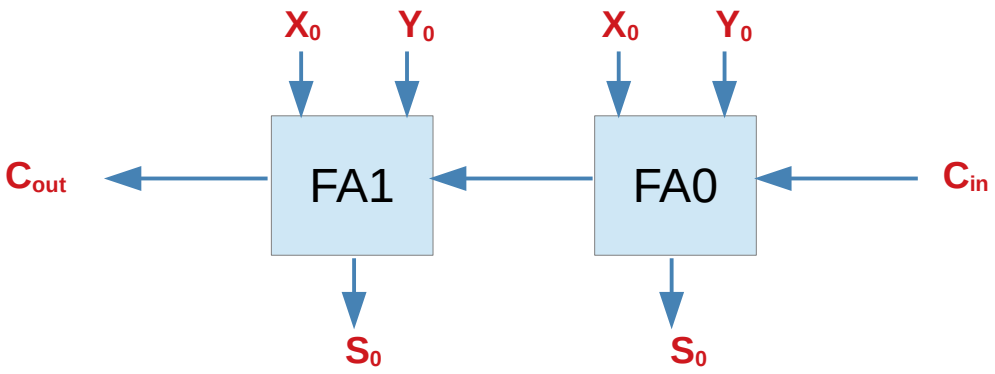


generated carry



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

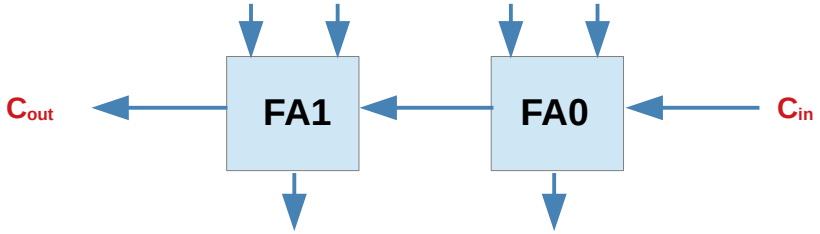
Critical path – all FA's in propagate mode

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

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

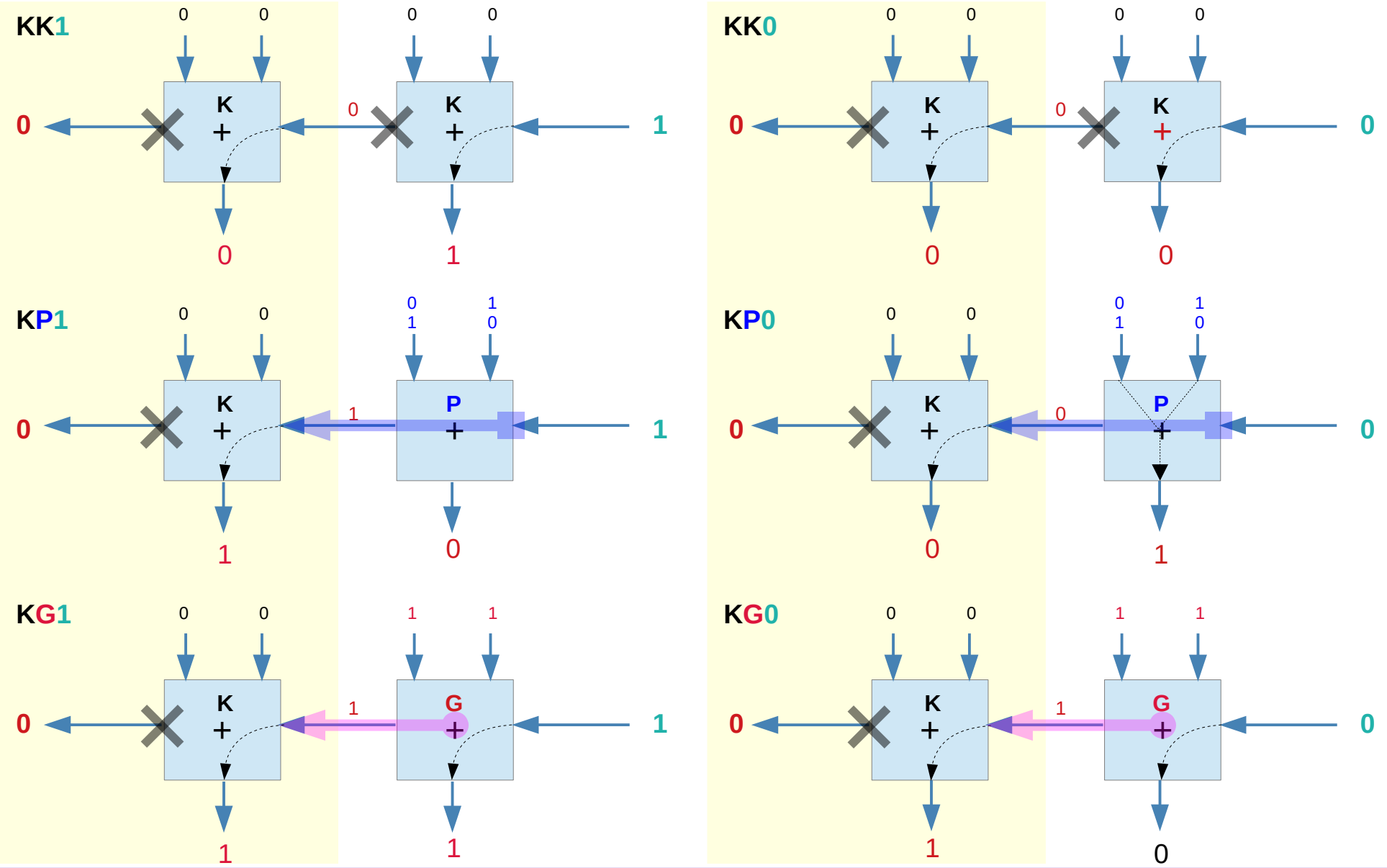
K, P, and G conditions in a 2-bit adder (2)

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

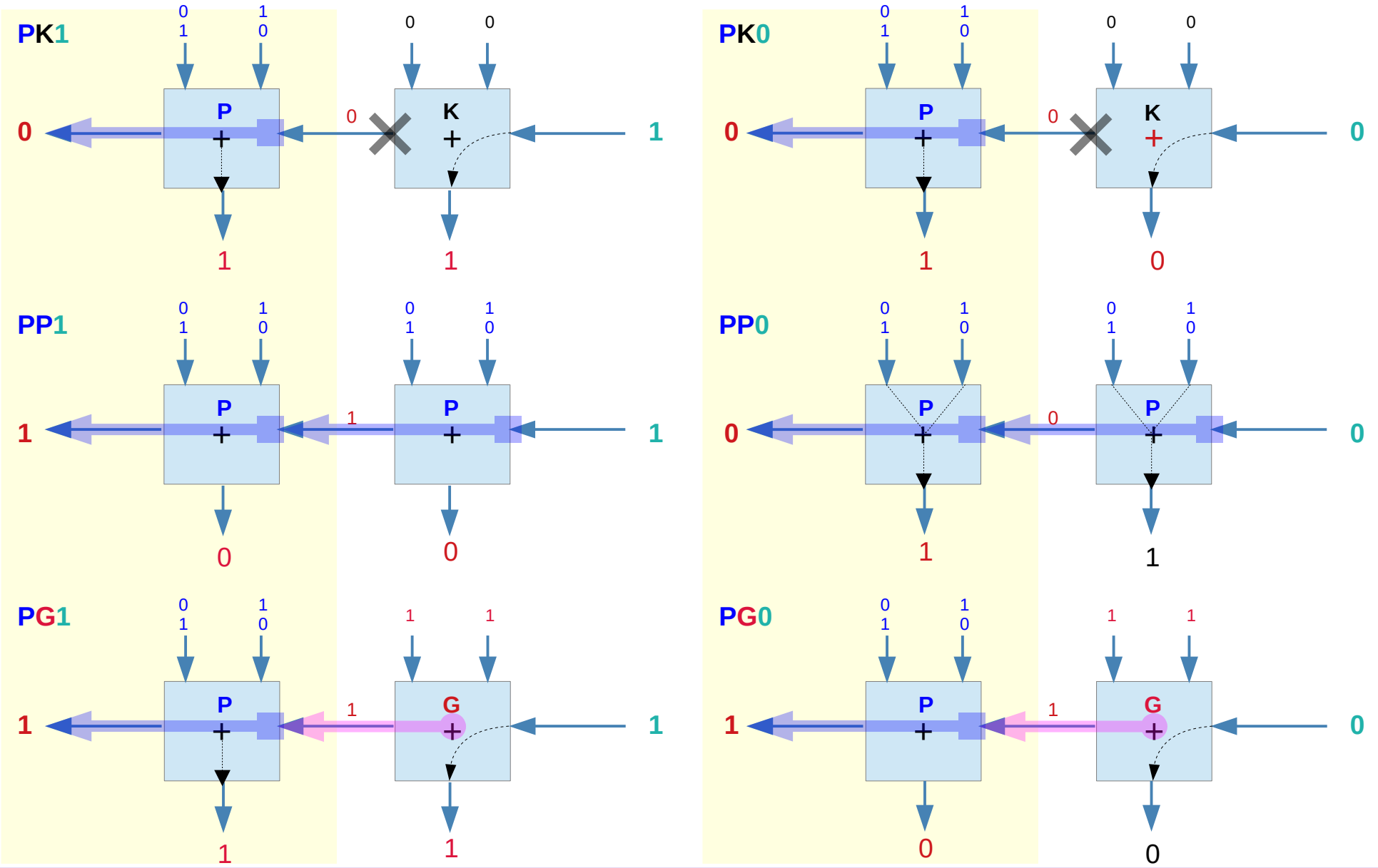


K	K	0
K	K	1
K	P	0
K	P	1
K	G	0
K	G	1
P	K	0
P	K	1
P	P	0
P	P	1
P	G	0
P	G	1
G	K	0
G	K	1
G	P	0
G	P	1
G	G	0
G	G	1

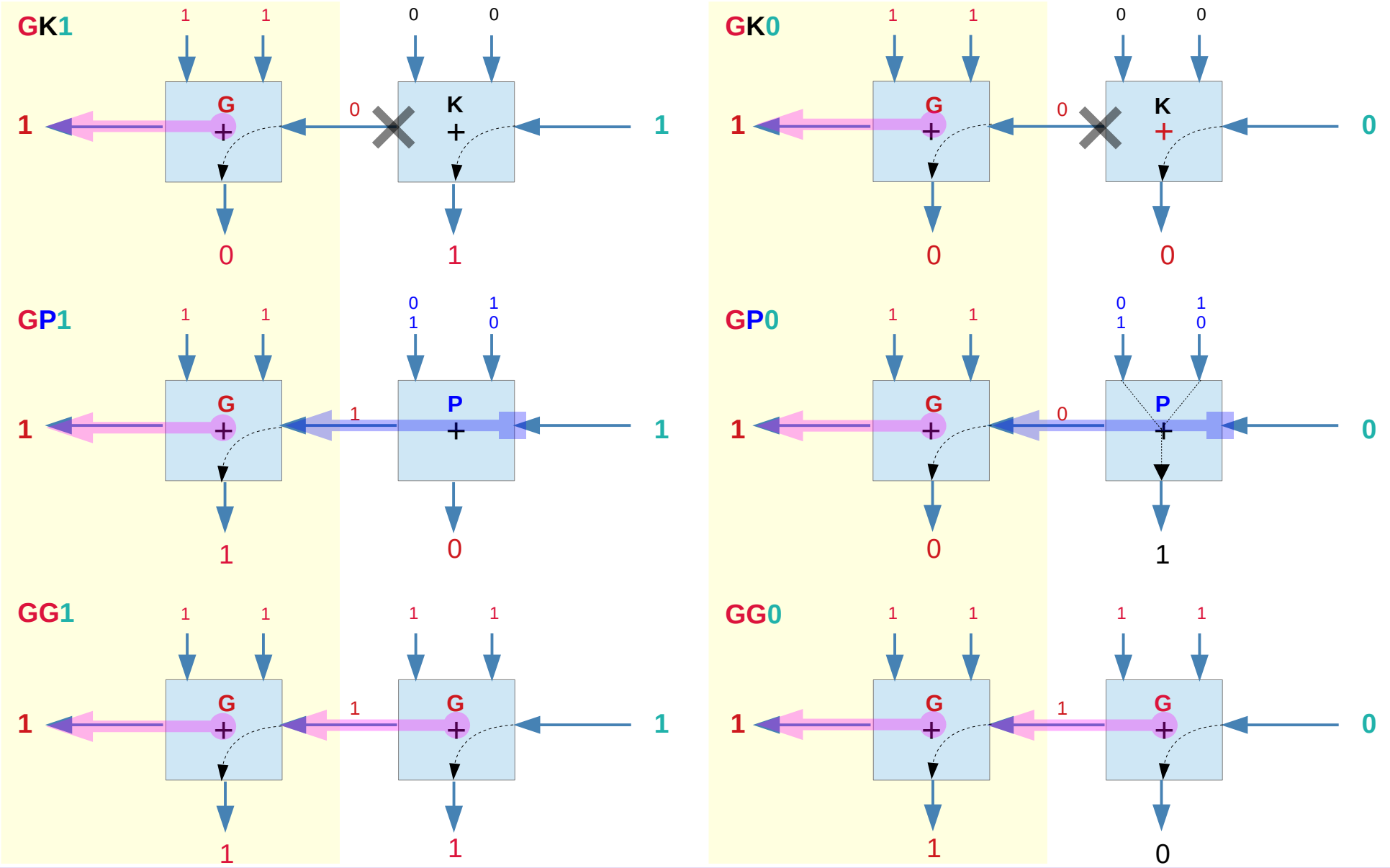
1. Cases when **FA1** is in the **K** mode



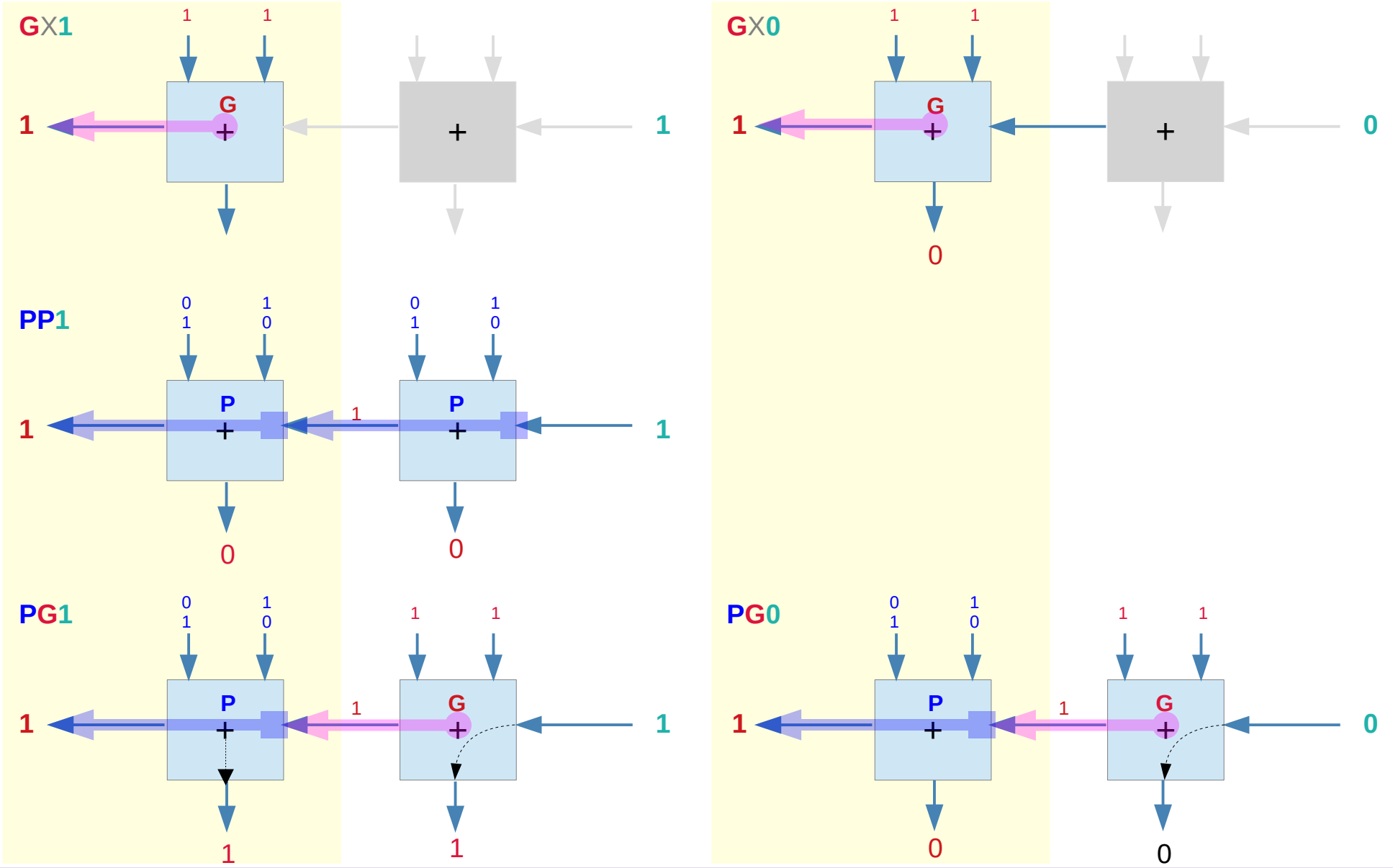
2. Cases when **FA1** is in the **P** mode



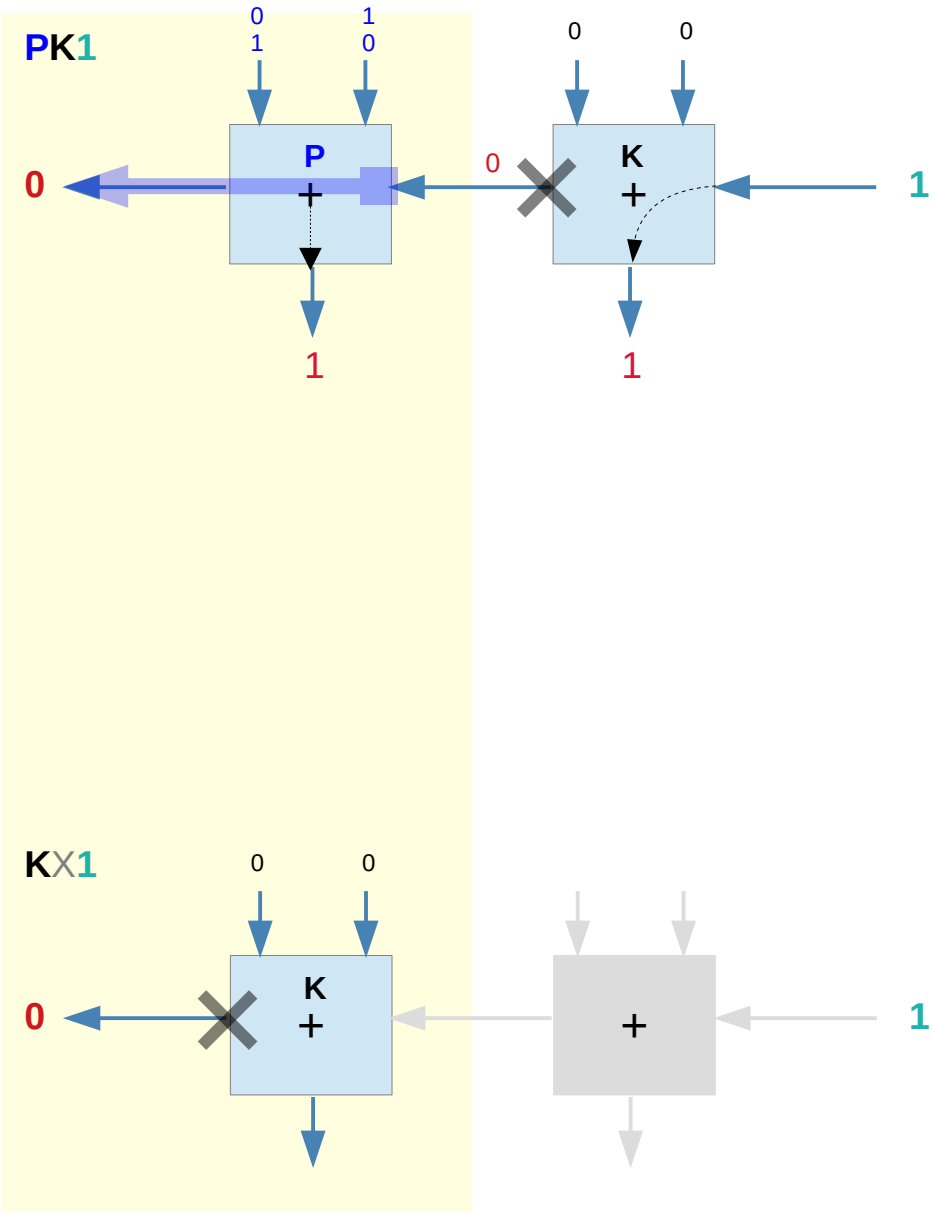
3. Cases when **FA1** is in the **G** mode

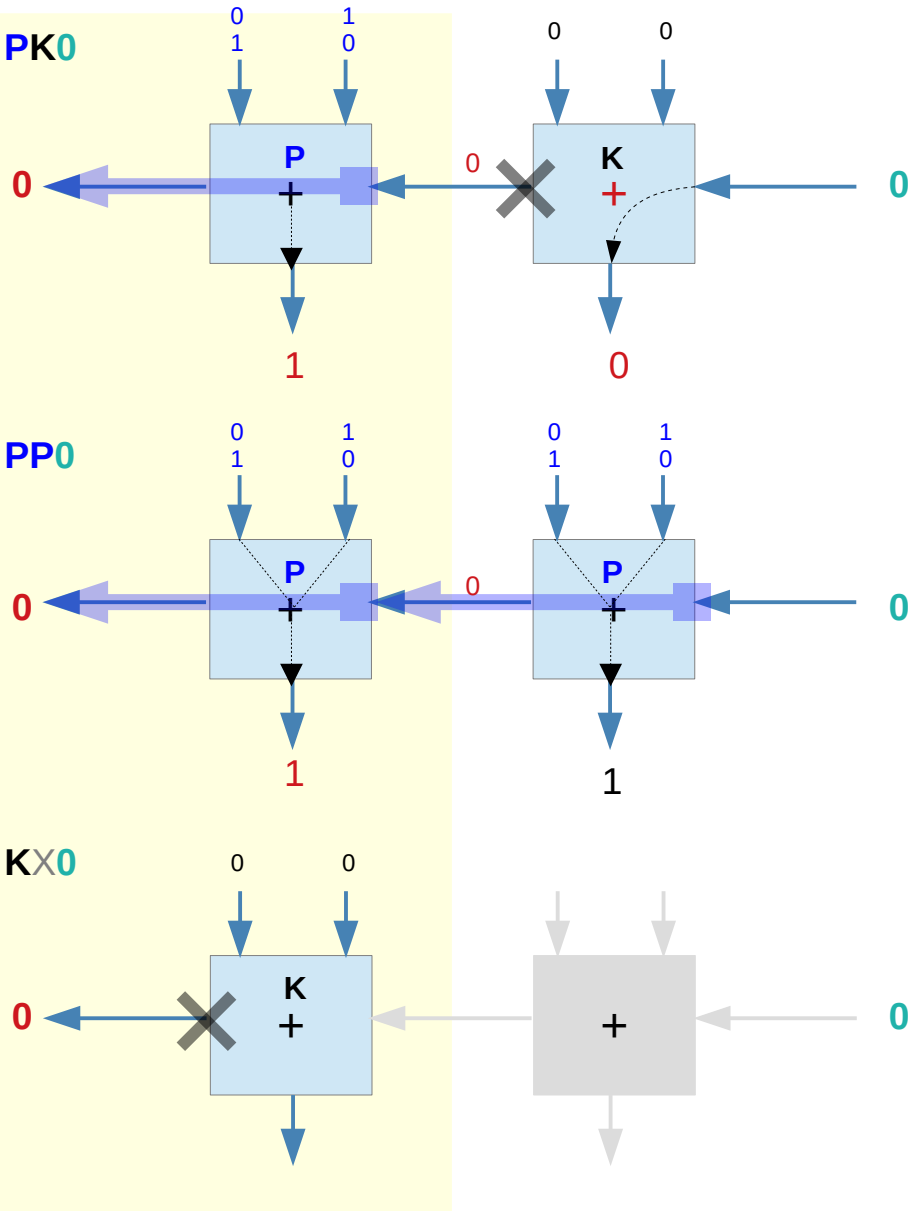


Cases for $C_{out} = 1$

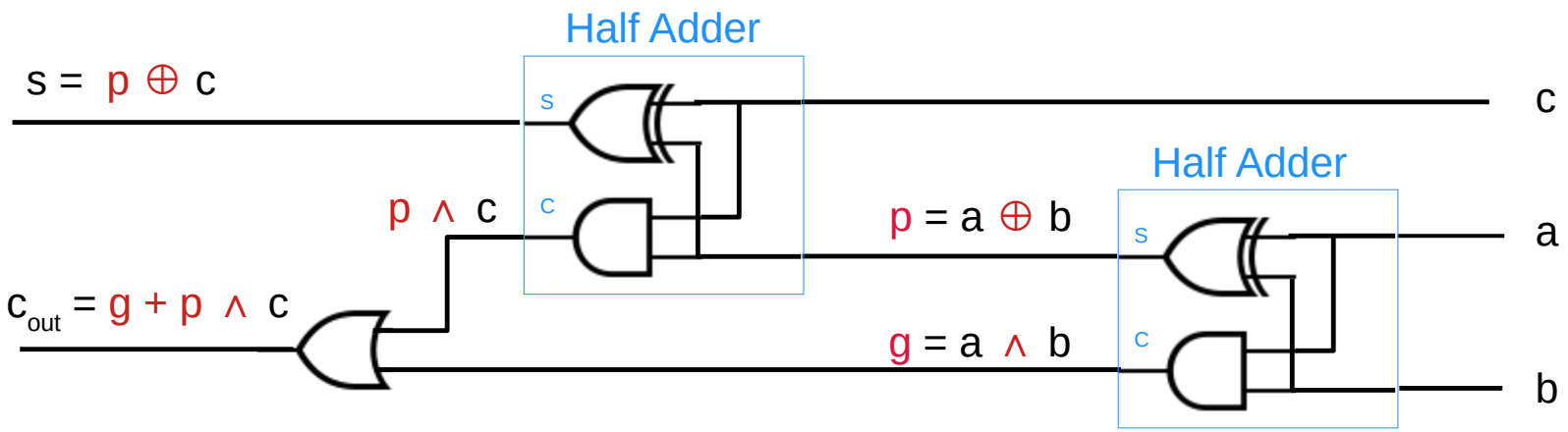


Cases for $C_{out} = 0$



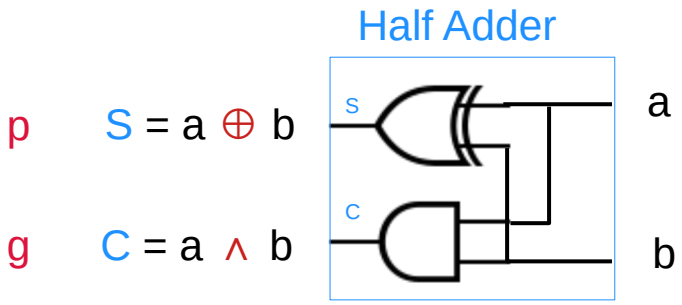


FA with P & G



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



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

Full adder with additional generate and propagate signals.

Ripple Carry Adder

$p_i = a_i \oplus b_i$

$g_i = a_i \wedge b_i$

$c_1 = g_0 + p_0 \wedge c_0$

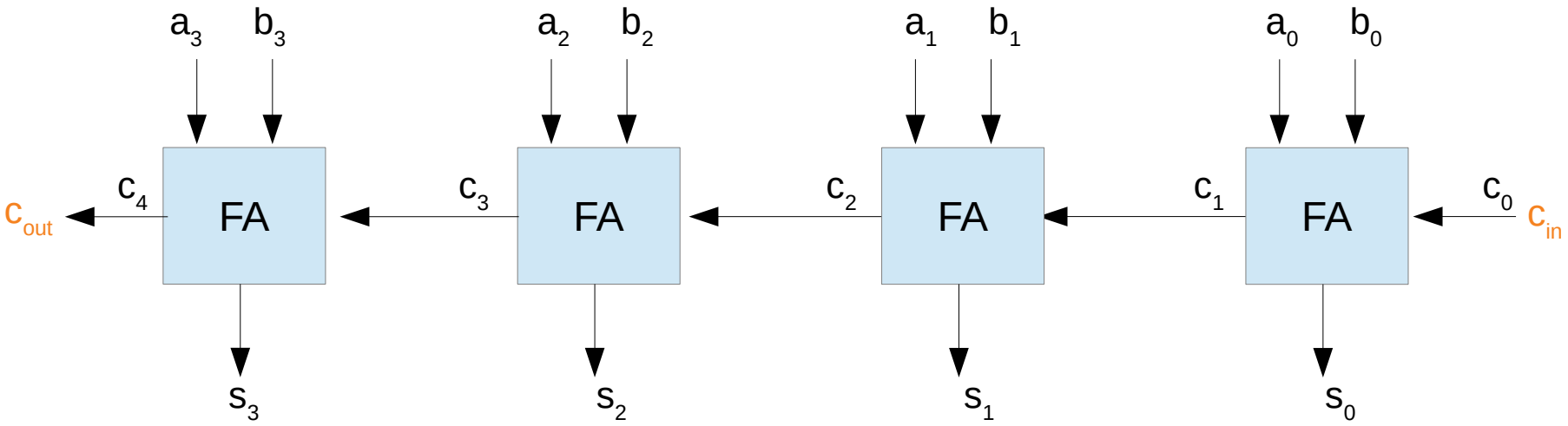
$c_2 = g_1 + p_1 \wedge c_1$

$c_3 = g_2 + p_2 \wedge c_2$

$c_4 = g_3 + p_3 \wedge c_3$

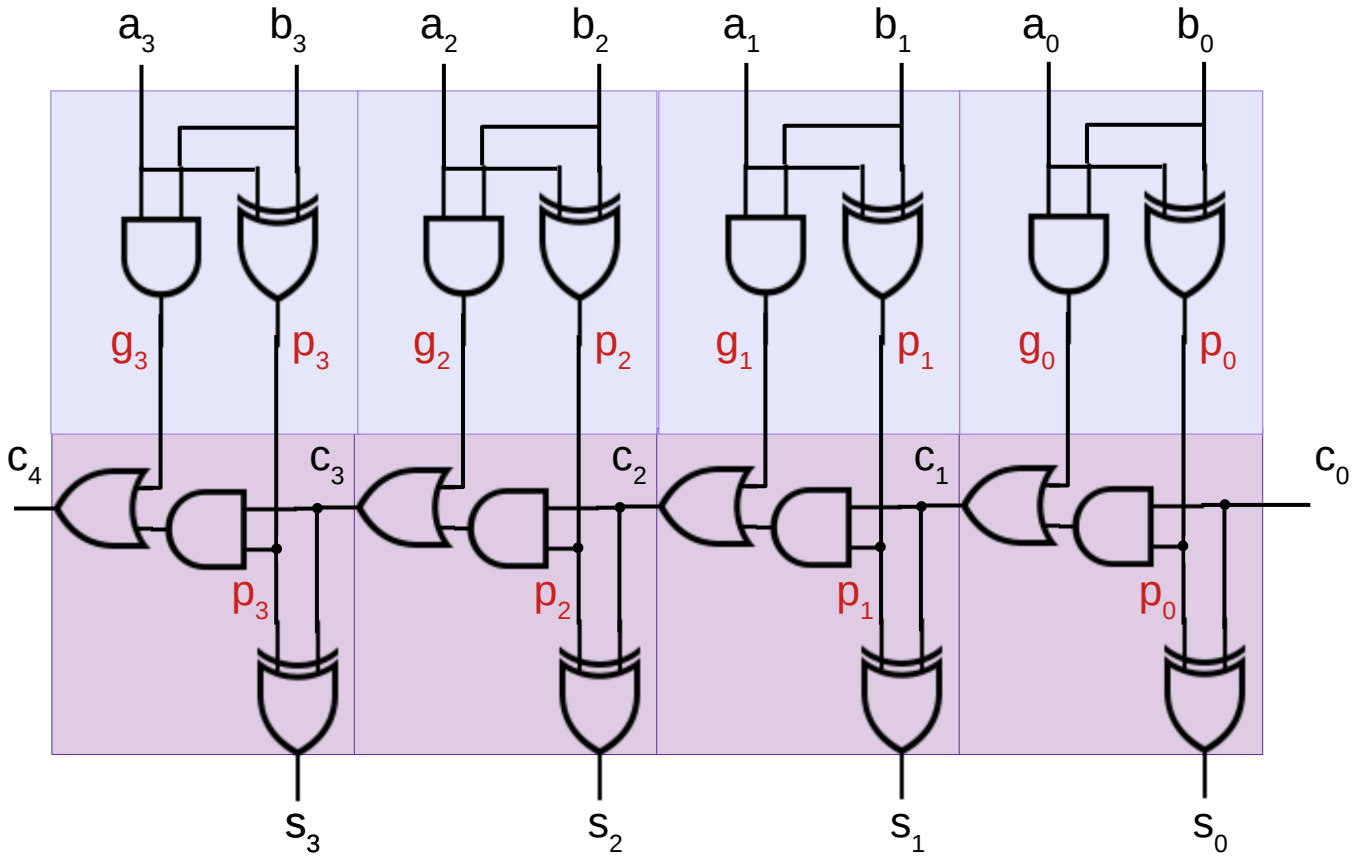
generated carry

propagated carry



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

4-bit Full Adder with P and G



Half Adder

$p_i = a_i \oplus b_i$

$g_i = a_i \wedge b_i$

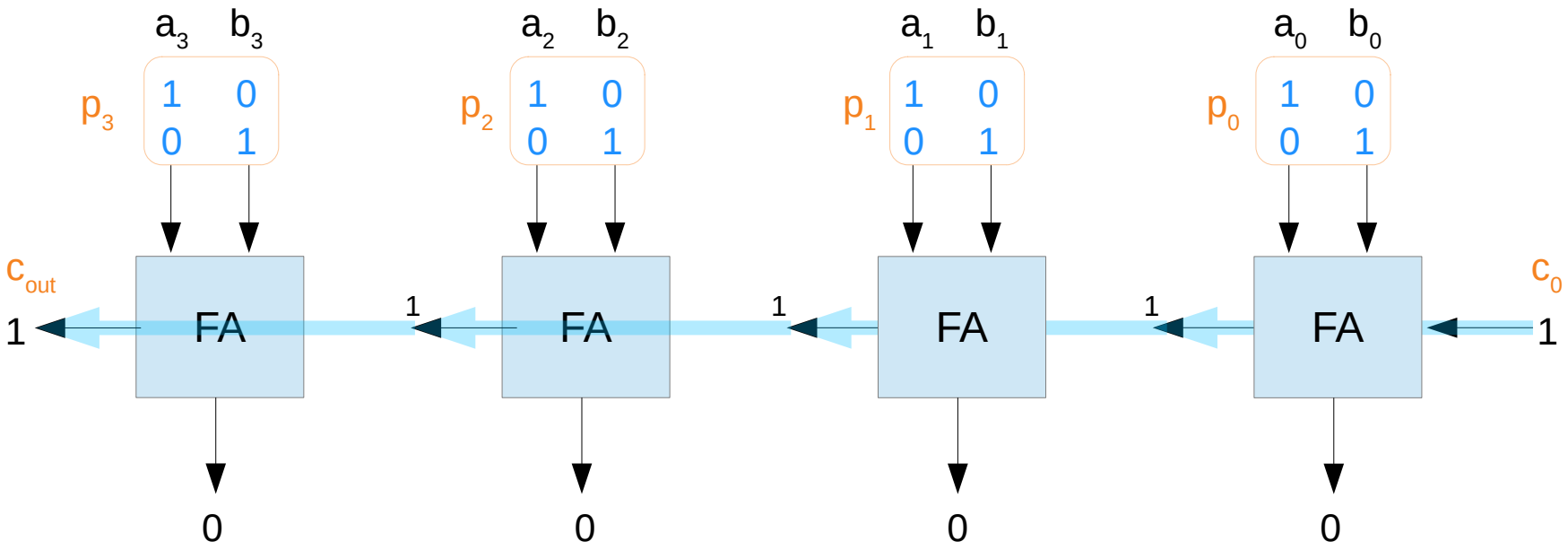
Half Adder

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

$s_i = p_i \oplus c_i$

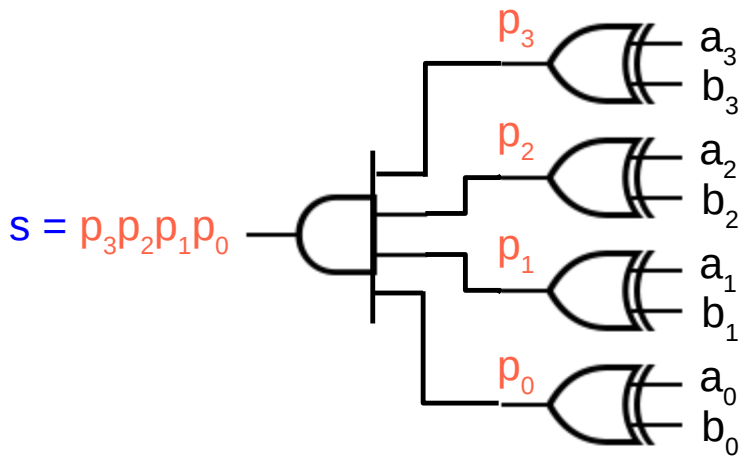
https://upload.wikimedia.org/wikiversity/en/1/18/RCA.Note.H.1.20151215.pdf

C_0 propagation condition



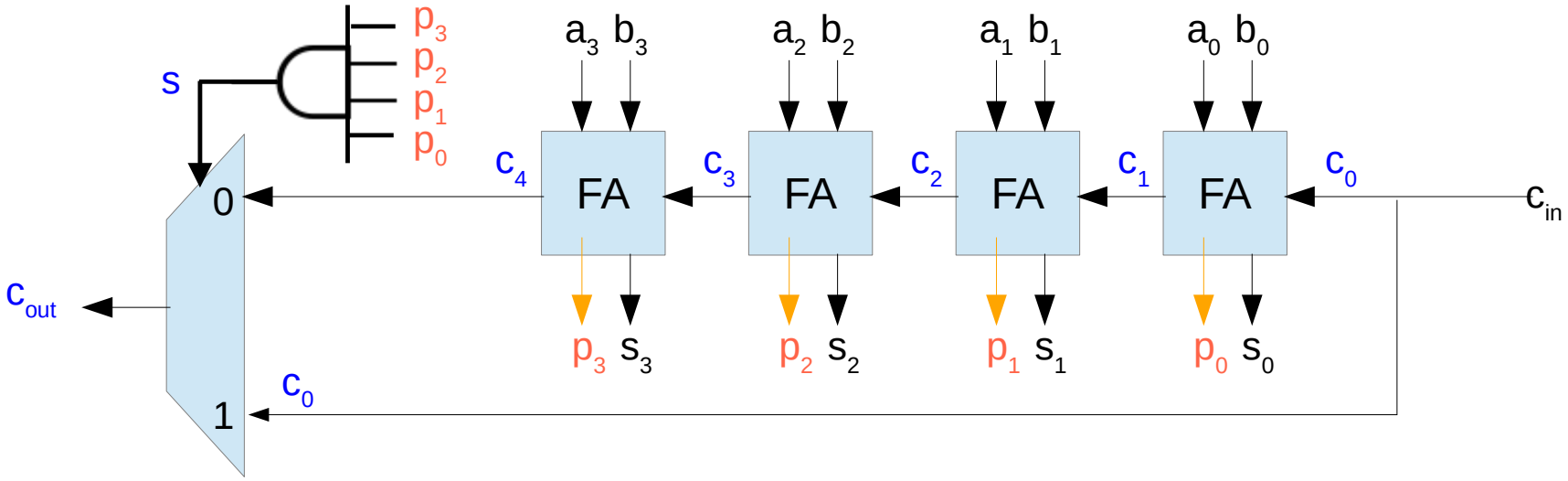
c_0 can be propagated to c_{out} only when $s = 1$

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



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

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**.

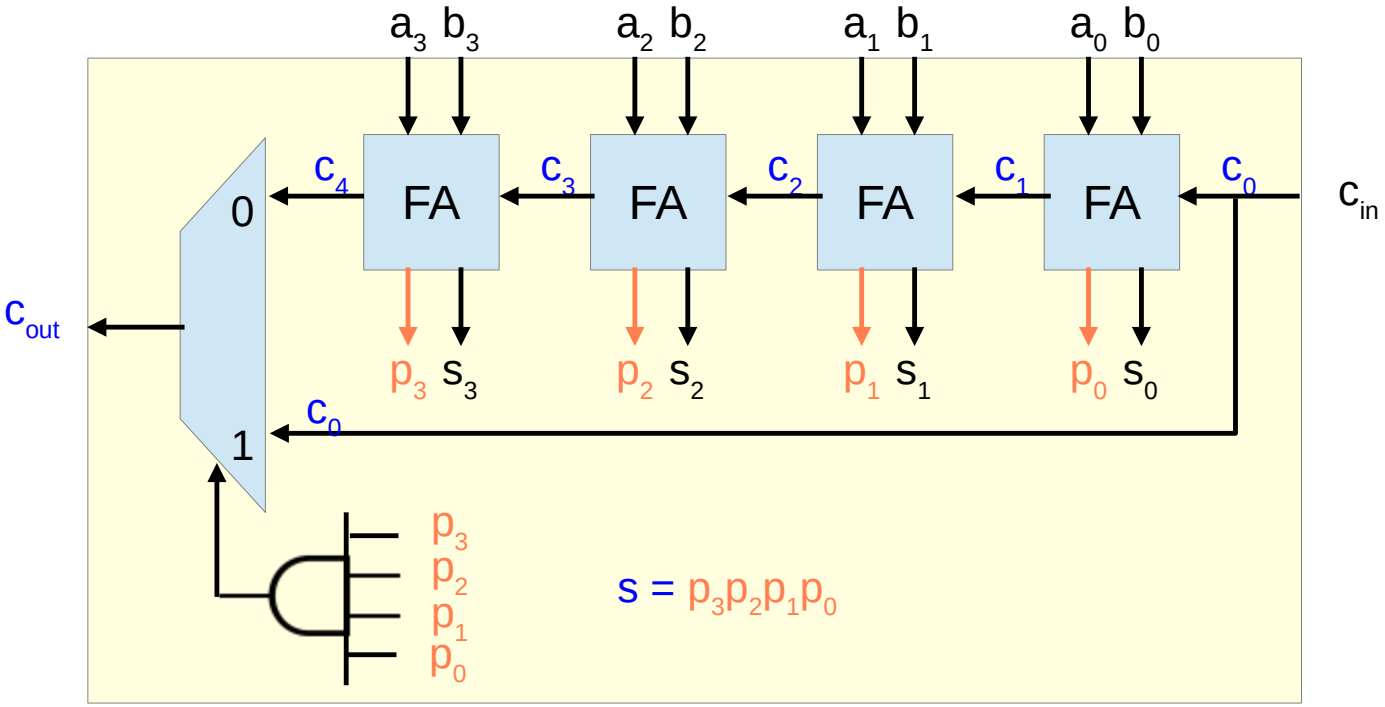
a multiplexer switches
either the last carry-bit c_n or the carry-in c_0
to the carry-out signal c_{out}

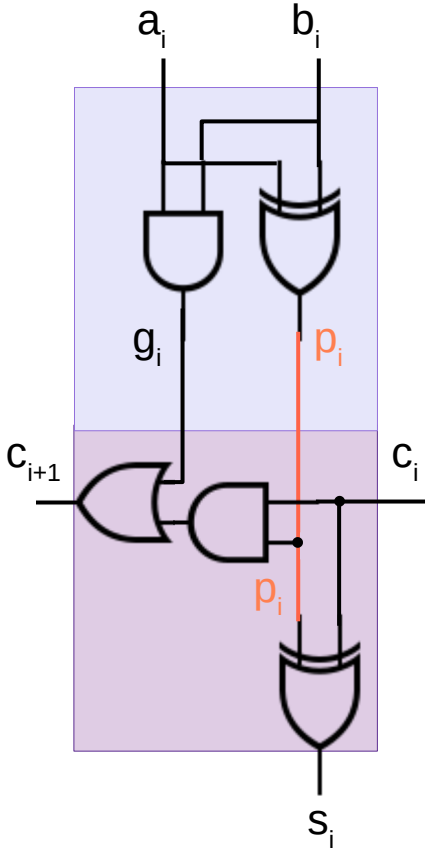
$s = p_3 \wedge p_2 \wedge p_1 \wedge p_0 = p_{[3:0]}$

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

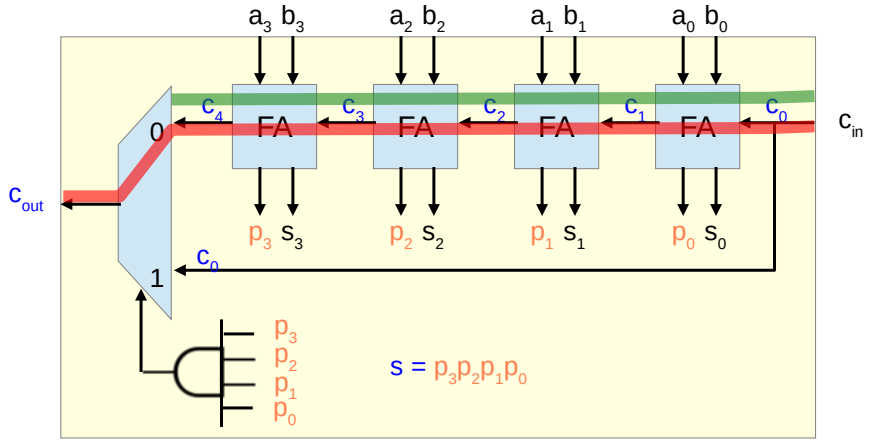
The critical path of a Carry Skip Adder begins at the first full adder, passes through all adders and ends at the sum bit s_{n-1}

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

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

$T_{RCA}(n)$

$T_{CSA}(n)$



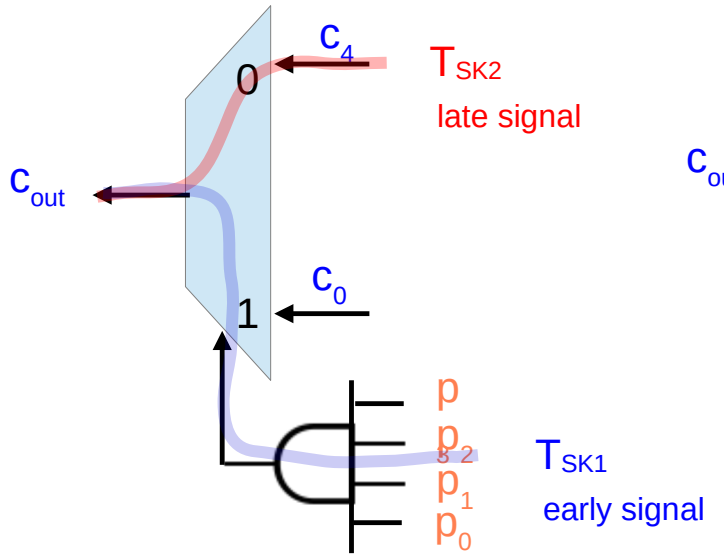
https://en.wikipedia.org/wiki/Carry-skip_adder

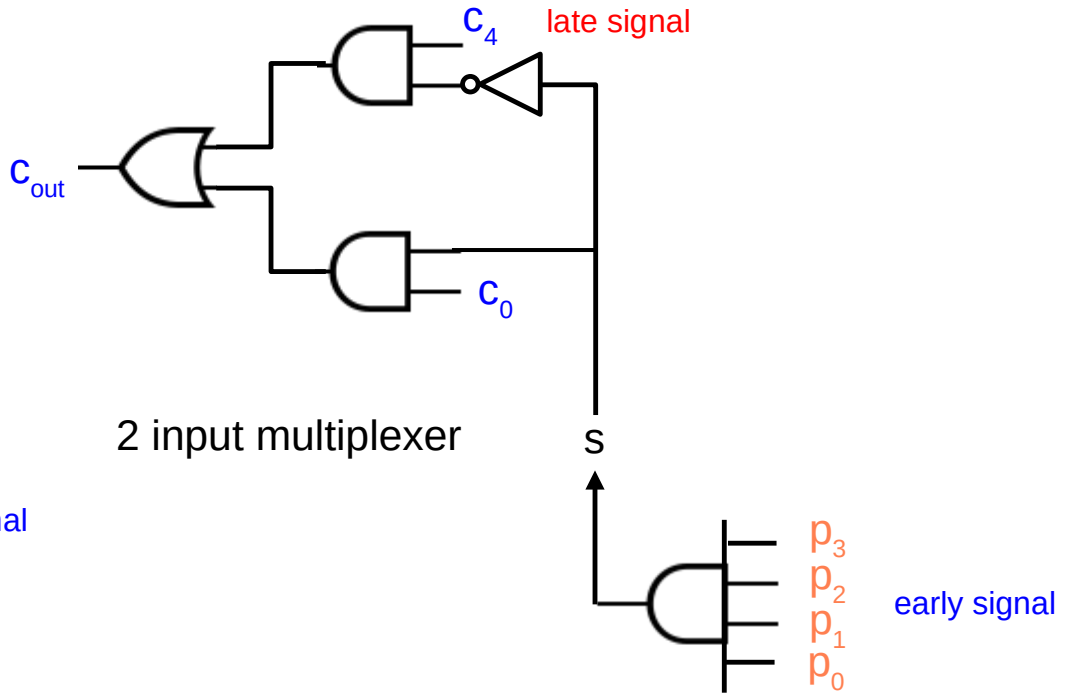
Carry Skip Adder

the skip logic 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

Carry Skip Adder

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

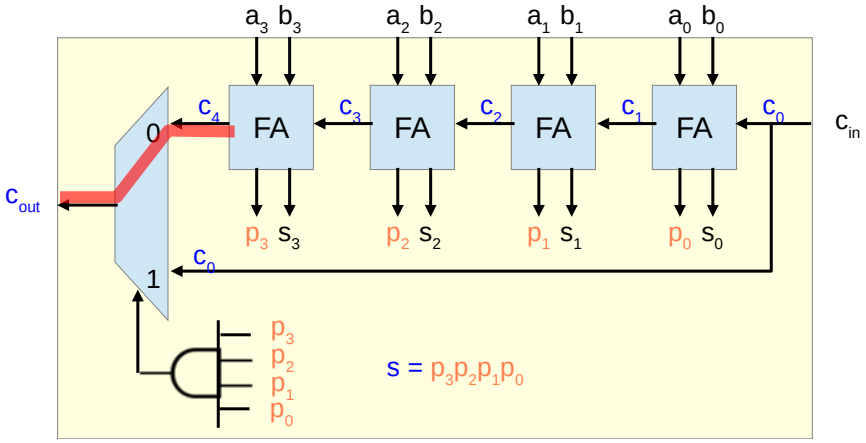
$p_i = a_i \oplus b_i$

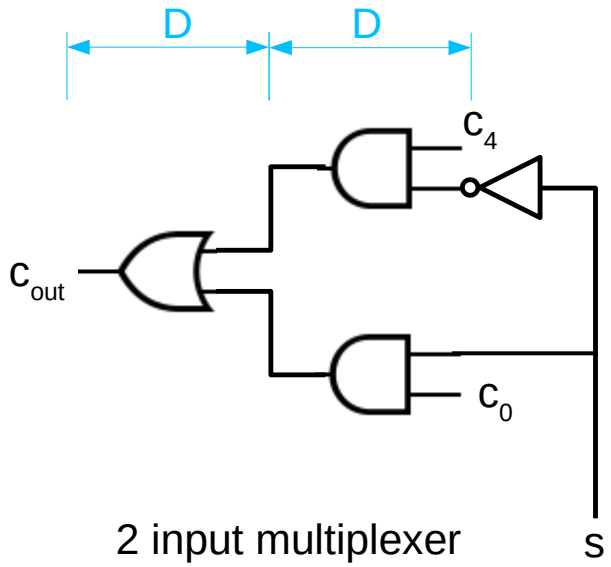
The critical path in a Carry Skip Adder consists of ripple carry path and mux path for ripple carry (T_{SK2})

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

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





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

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.

Fixed-size block CSA (FCSA)

number of blocks

bits per block
block size

$$n = R \cdot k$$

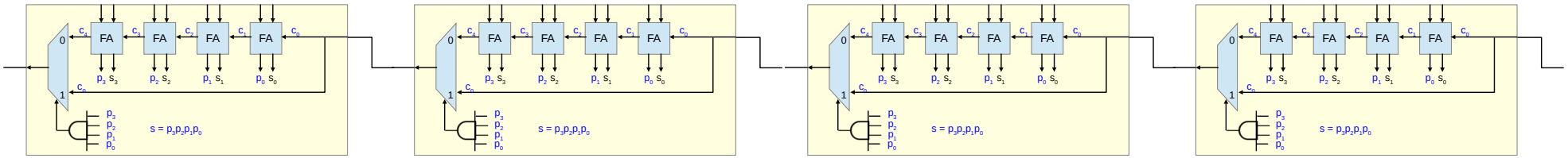
R blocks

k bits
[(R)k-1 … (R-1)k]
= [N-1 … N-k-1]

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

k bits
[(2)k-1 … (1)k]
= [2k-1 … k]

k bits
[(1)k-1 … (0)k]
= [k-1 … 0]



Www.cs.tufts.edu

Carry Skip Adder

the critical path

the longest carry path must be
- generate in the first block
- terminated in the last block
- propagated 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

Fixed-size block CSA (FCSA)

R groups

k bits

$n = R \cdot k$

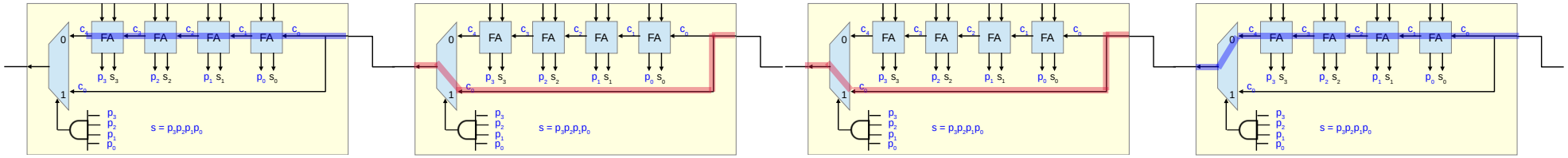
the last block

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

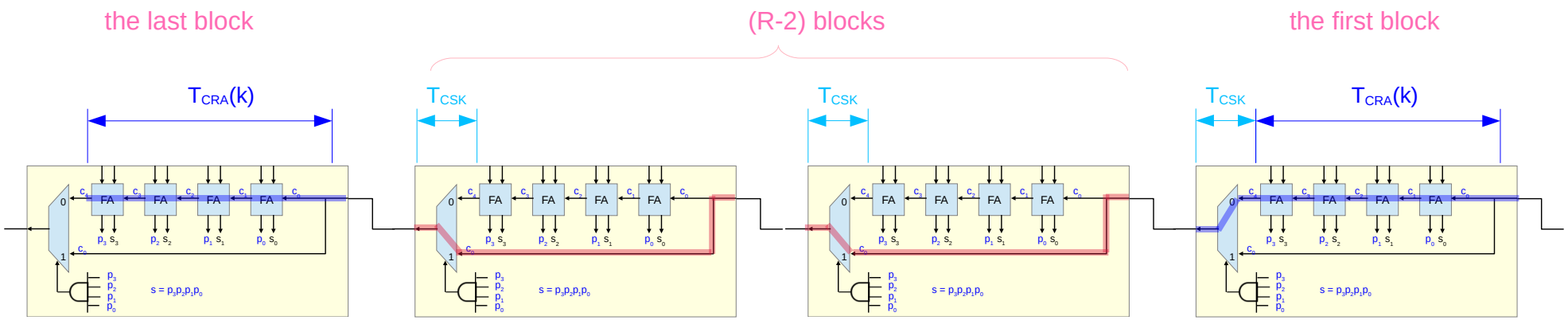
- the ripple path and the skip element of the first block
  $T_{CRA}(k)$ + T_{CSK} $T_{CRA[0:cout]}(k)$
- the skip paths that are enclosed between the first and the last block
  $(R-2)T_{CSK}$
- finally the ripple path of the last block
  $T_{CRA}(k)$

Fixed-size block CSA (FCSA)

R groups

k bits

$n = R \cdot k$



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

Carry Skip Adder

Consider a *n* bit adder, where *n* is a multiple of 4.

Each set of 4 neighboring stages can be considered
a block that can generate, propagate, or absorb a carry.

Divide the adder into *n*/4 such blocks and,
for each block, add a 4-input AND gate
to produce a group propagate signal.

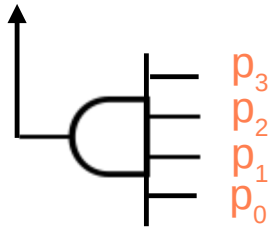
A carry signal entering a certain block can be propagated
past the block without waiting for the signal to propagate
through the 4 individual stages of the block.

Fixed-size block CSA (FCSA)

R groups

k bits

$$n = R \cdot k$$



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

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

Carry Skip Adder

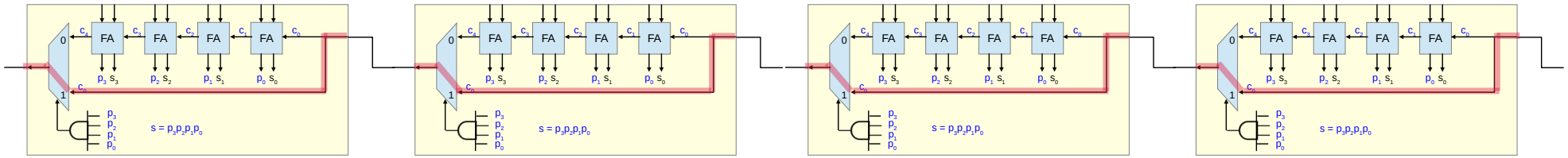
If all *n*/4 blocks propagate, a carry entering the least significant stage will pass to the most significant carry-out in time *n*/4 times the delay through the carry-skip unit.

Fixed-size block CSA (FCSA)

R groups

k bits

n = R·k



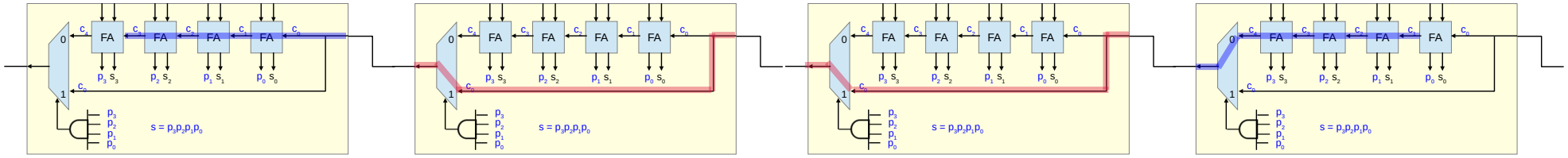
the last block

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

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

Carry Skip Adder

If any block generates a carry, that carry will propagate through the remaining stages of the block,
and then through the carry-skip gates to the final block,
where it may have to propagate through 3 stages
to reach the most significant adder.

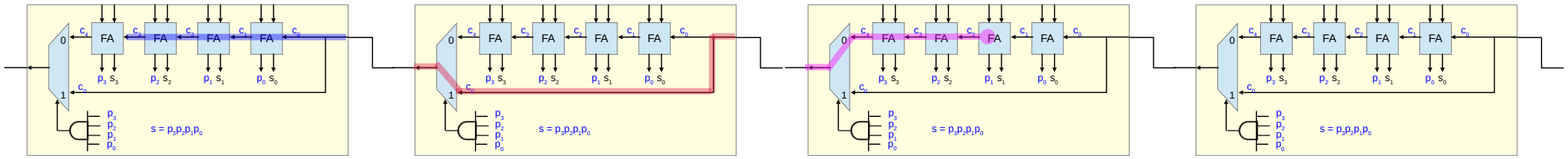
Fixed-size block CSA (FCSA)

R groups

k bits

n = R·k

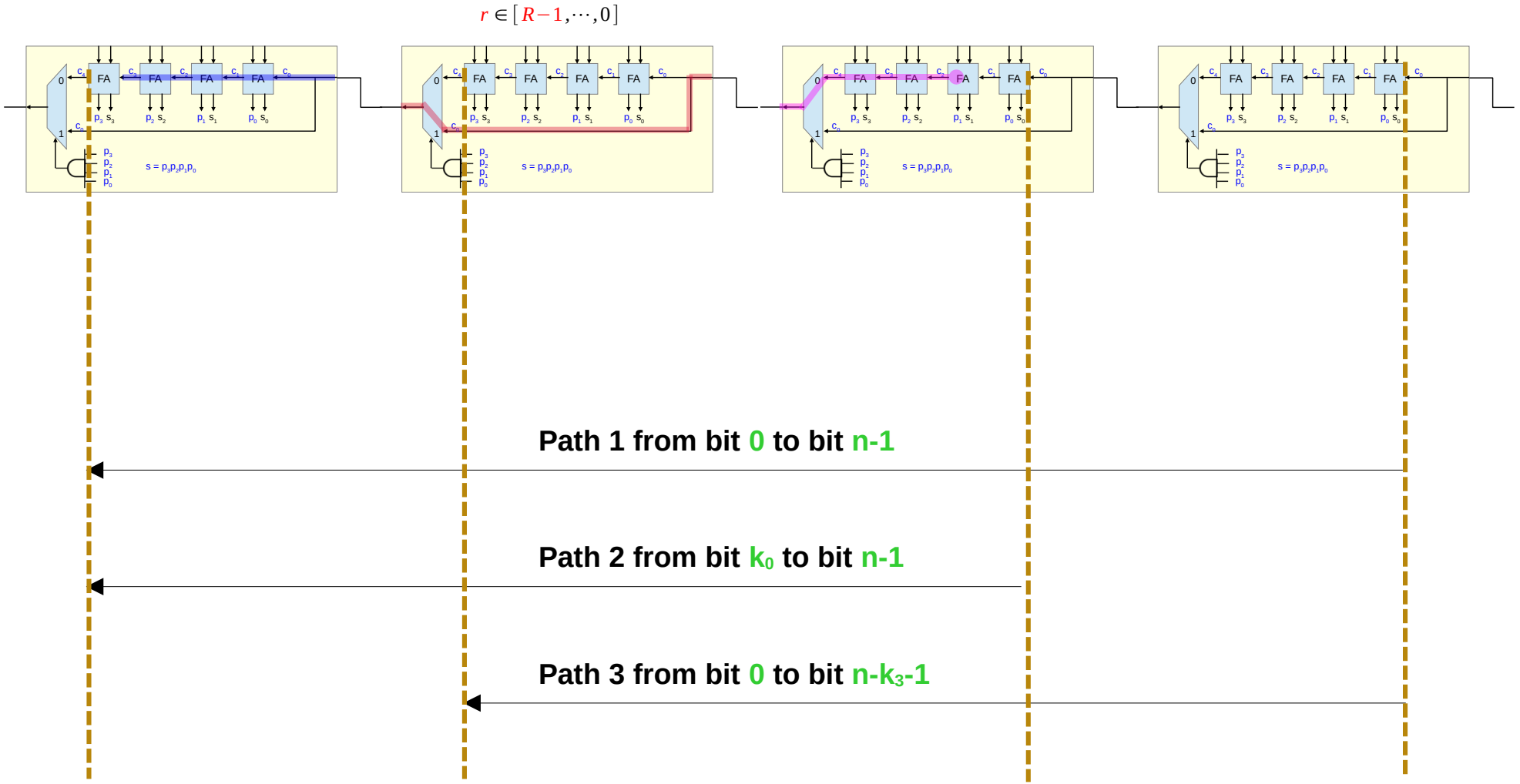
$r \in [R-1, \cdots, 0]$



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

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

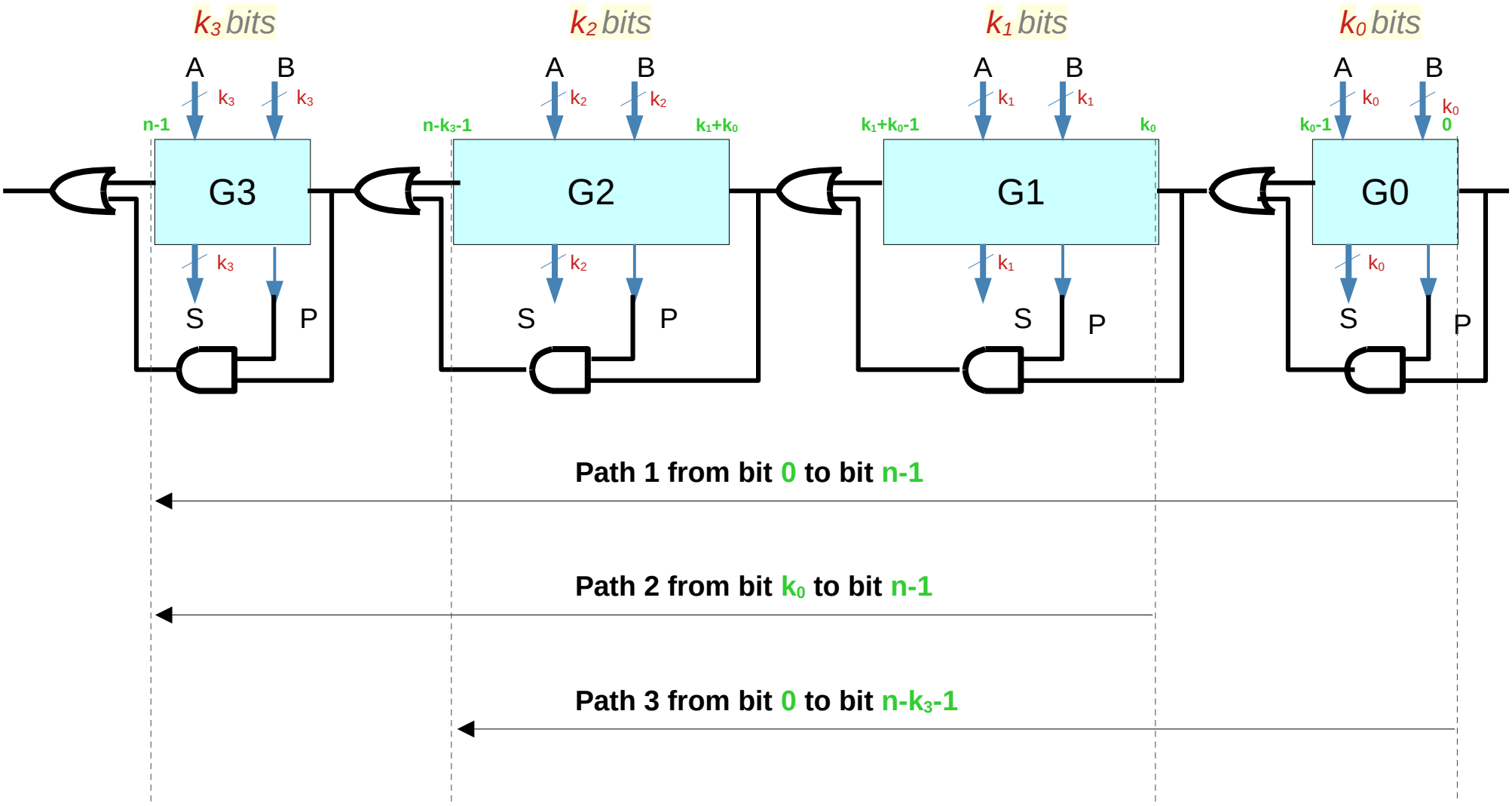
Variable size Block Carry Skip Adder (3)



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)



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

The longest delay path begins with a carry generated in stage 0
in the least significant block, propagates through 3 stages
in that block, then through the OR gate,
then through *n* - 2 carry-skip units,
and then through 3 of the 4 stages in the most significant block,
to the c_{n-1} signal. We can generalize these results

for a block size of R in a *n* bit adder as follows:

Fixed-size block CSA (FCSA)

R groups

k bits

$$n = R \cdot k$$

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

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

Carry Skip Adder

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

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

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

Fixed-size block CSA (FCSA)

R groups

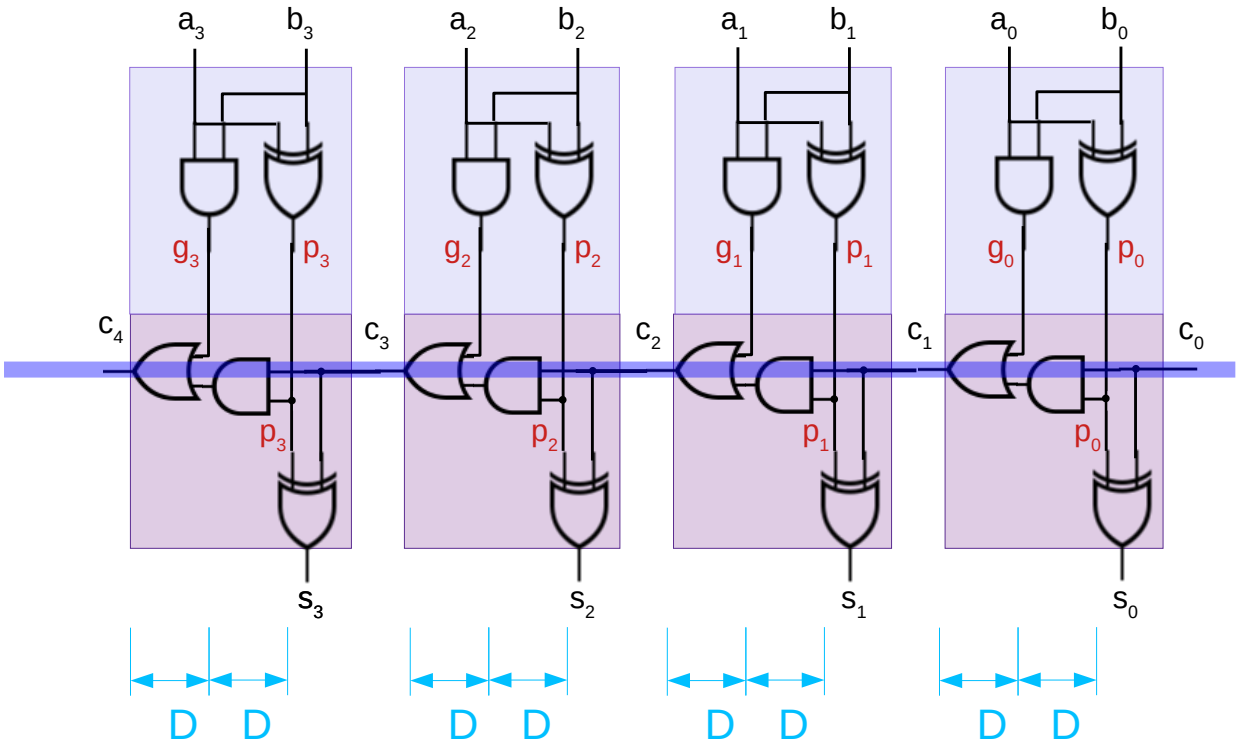
k bits

$$n = R \cdot k$$

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

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

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 ripple path and the skip element of the first block $T_{CRA}(k) + T_{CSK}$
- 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 $T_{CRA}(k)$

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

R groups

k bits

$$n = R \cdot k$$

$$= k\,2D + 3D + (R-1)2D + (k+2)2D$$
$$= 3D + k2D + R2D - 2D + k2D + 4D$$
$$= (2k+R)2D + 5D$$

Optimal block size k

Fixed-size block CSA (FCSA)

R groups

k bits

n = R·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

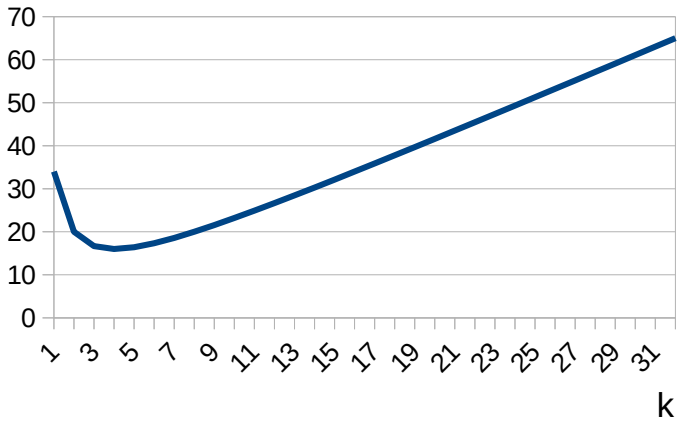
https://en.wikipedia.org/wiki/Carry-skip_adder

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)



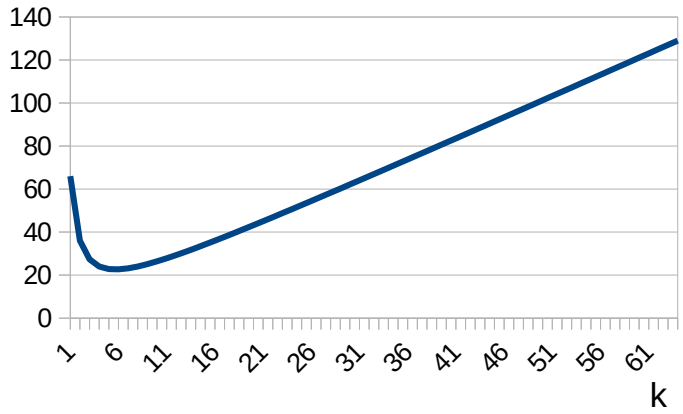
$4 = \sqrt{32/2}$ $\quad k_{opt} = \sqrt{\frac{n}{2}}$

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)



$5.6 = \sqrt{64/2}$ $\quad k_{opt} = \sqrt{\frac{n}{2}}$

n = 64bits
→ k = 6

Fixed-size block CSA (FCSA)

R groups

k bits

n = R·k

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

Asymptotic Analysis

Fixed-size block CSA (FCSA)

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

The optimal block size k for a given adder width n

$k = \sqrt{n/2}$

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

$= (\sqrt{2n} + \sqrt{n^2 / (n/2)})\ 2D$

$= (\sqrt{2n} + \sqrt{2n})2D$

$= (2\sqrt{2n})2D$

R groups

k bits

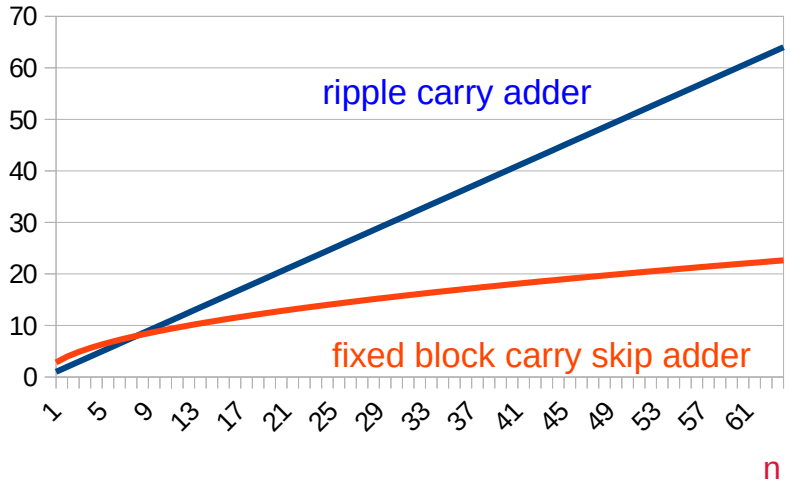
n = R·k

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

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

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

$$= (2\sqrt{2n})2D \qquad when \quad k_{opt} = \sqrt{\frac{n}{2}}$$

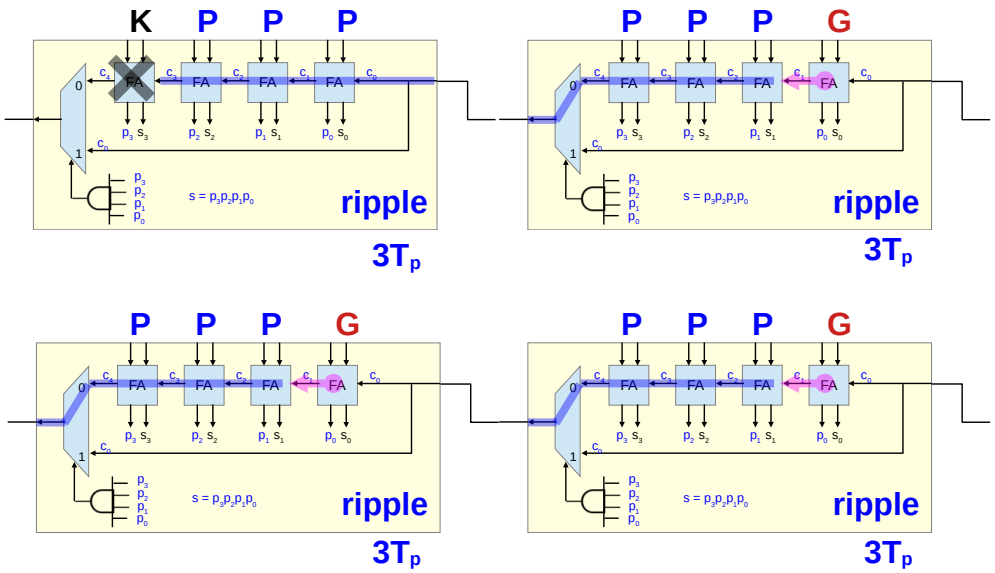


https://en.wikipedia.org/wiki/Carry-skip_adder

Carry Skip Adder

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

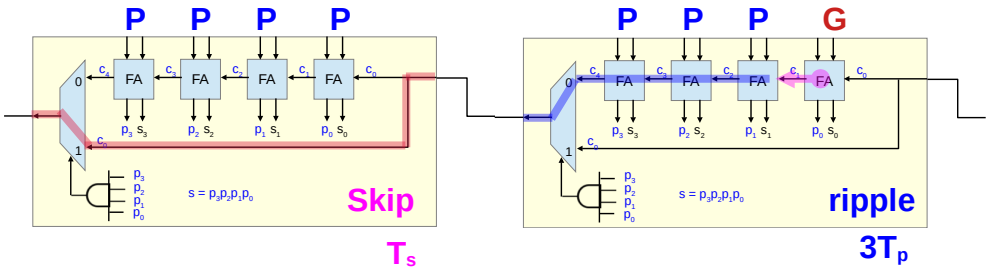
however, if the second block generates a carry itself,
or kill the carry, then that is the end of the critical path



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

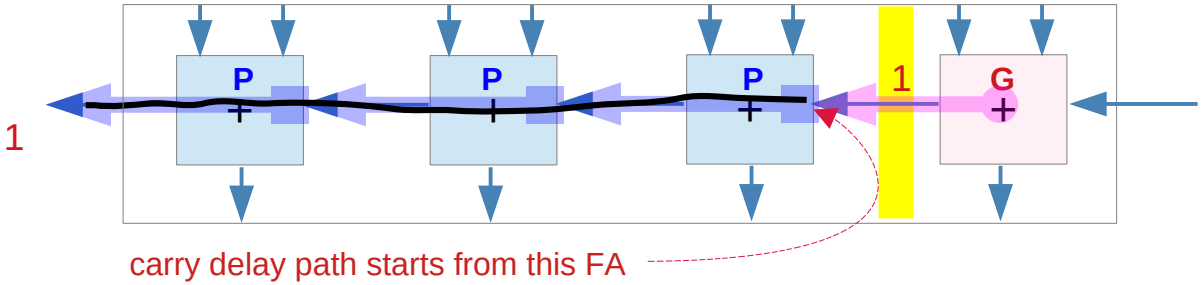


https::/electronics.stackexchange.com/questions/21251/critical-path-for-carry-skip-adder

Critical Carry Path (1)

$T_s < 3T_p$

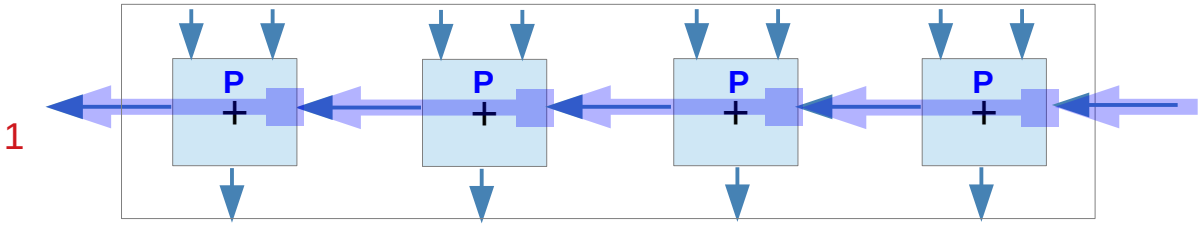
For longest carry path, if any block generates a carry,
that carry will propagate through the remaining 3 FA's of that block



Least Significant Block

C_{in} = 1 or 0

and then through the carry skip gates to the final block,



Middle Blocks

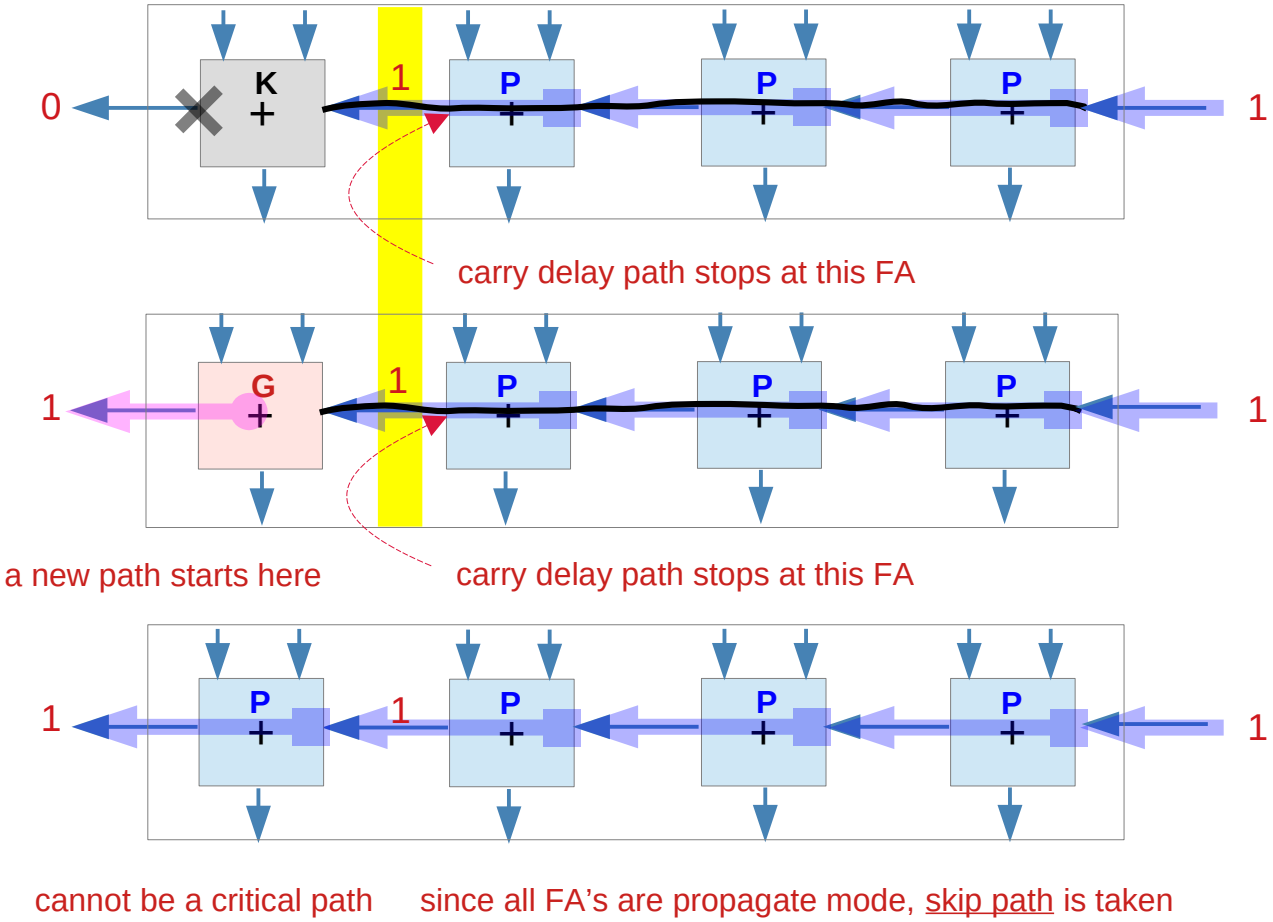
1

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 final block, it may have to propagate 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

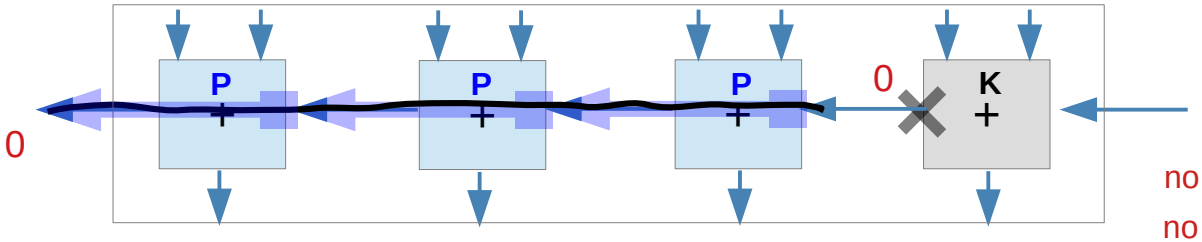
$T_s < 3T_p$

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

Critical Carry Path (3)

$T_s < 3T_p$

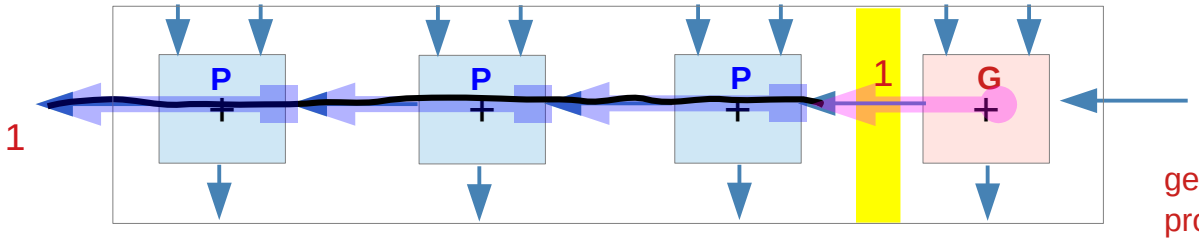
Cases for the Least Significant Block



1/0

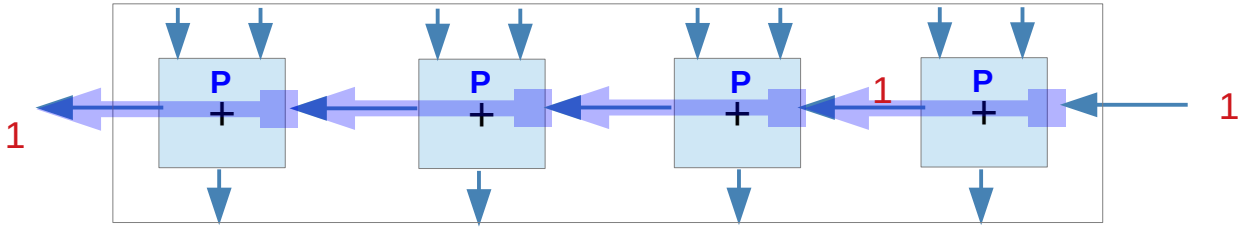
no carry out

no carry propagation



1/0

generated carry can be
propagated to other blocks



1

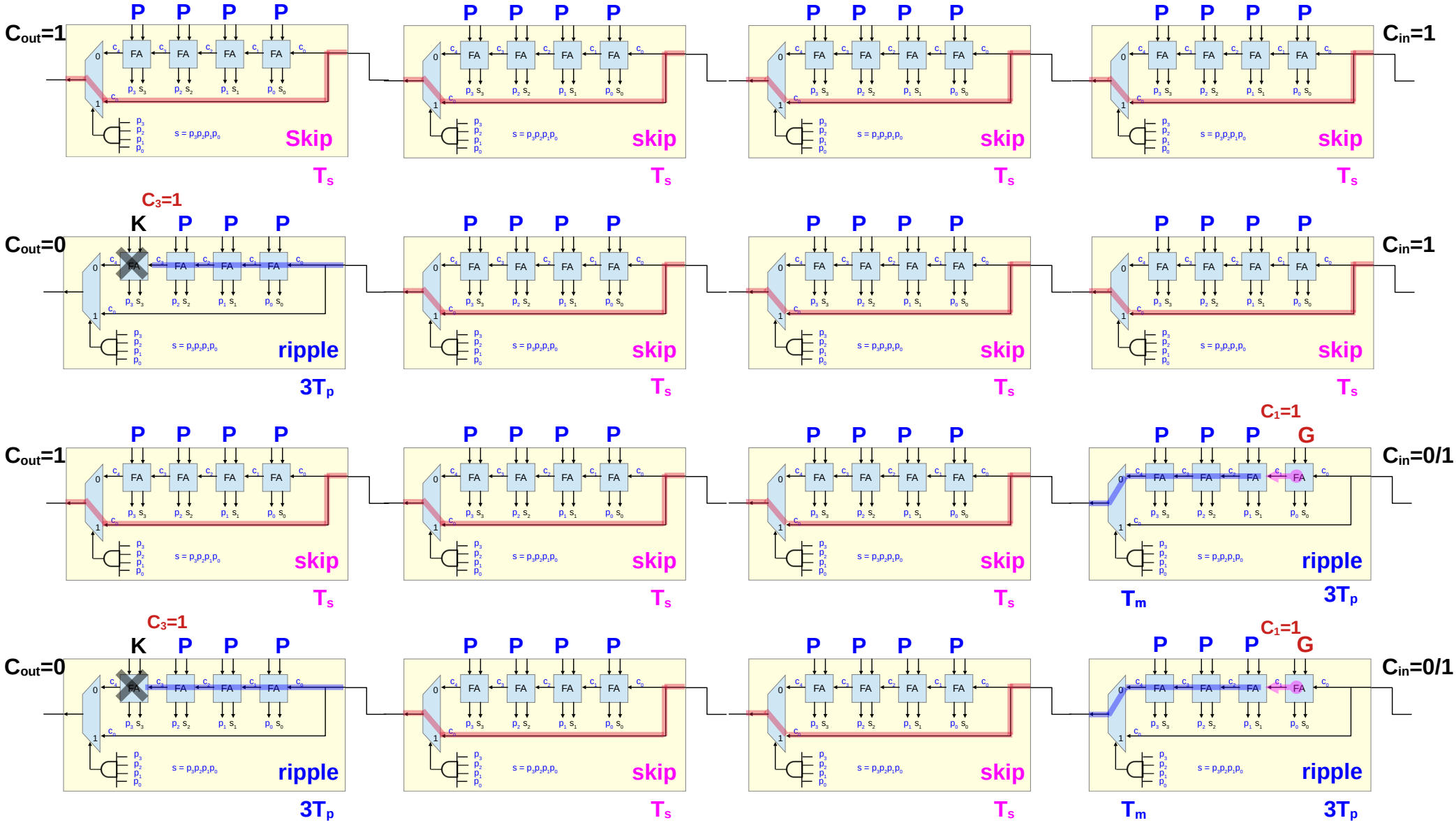
$T_s < 3T_p$

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

Fixed-size block CSA (FCSA)

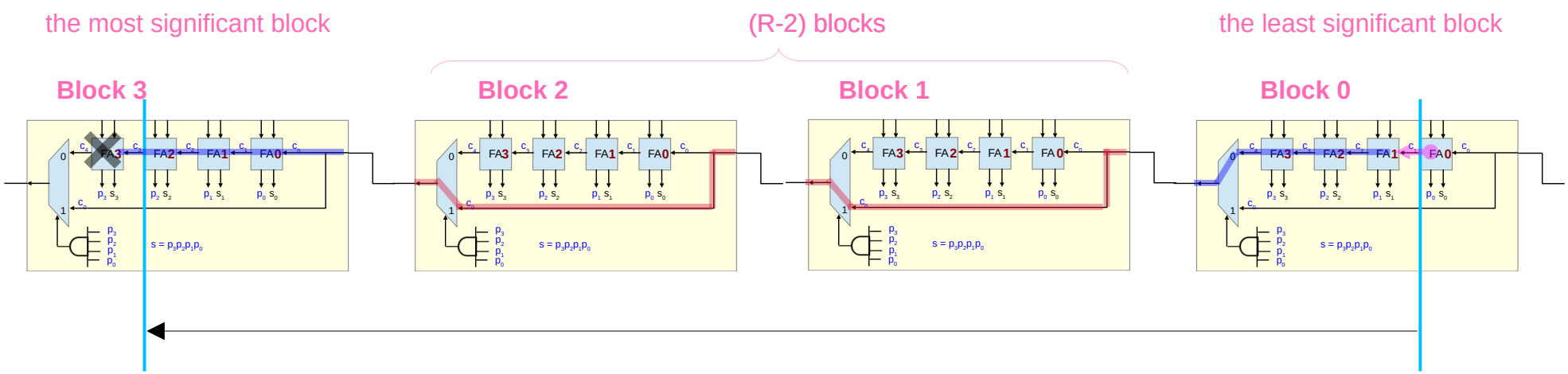
R groups

k bits

$n = R \cdot k$

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



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})

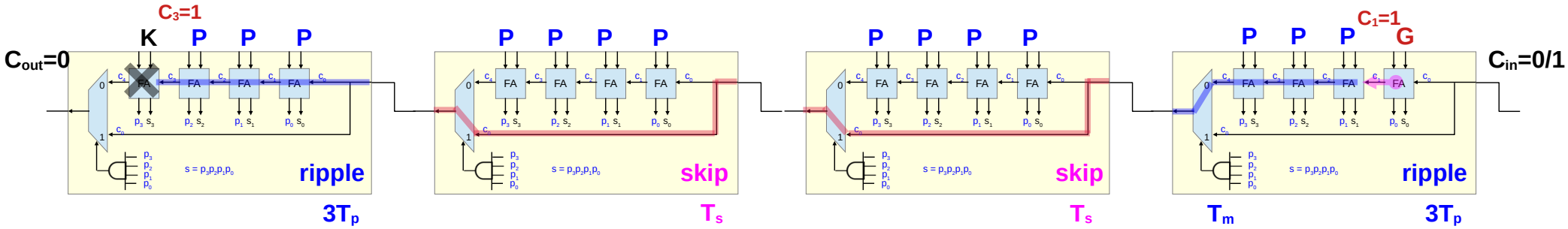
T_s is the delay through one carry-skip stage

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

Fixed-size block CSA (FCSA)

R groups

k bits

n = R·k

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*

$$Tfixed = (b-1)T_p + D + (k/b-2)T_s + (b-1)T_p$$

The original formula in the literature

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

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·k

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.

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

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

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

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

Carry Skip Adder

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

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

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

$= 4D - 2nD/k^2 = 0$

$4D = 2nD/k^2$

$k^2 = n/2$

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

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

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

Carry Skip Adder

Fixed-size block CSA (FCSA)

R groups

k bits

$n = R \cdot k$

The corresponding adder delay is:

$T^{opt}_{fixed} = 4Dk^{opt} + 2nD/k^{opt} - 7D$ $\qquad k^{opt} = \sqrt{n/2}$

$= 4D\sqrt{n/2} + 2nD/\sqrt{n/2} - 7D$

$= (4D\sqrt{n})/\sqrt{2} + 2\sqrt{2}nD/\sqrt{n}) - 7D$

$= (4/\sqrt{2}D\sqrt{n} + 2\sqrt{2}D\sqrt{n}) - 7D$

$= (4/\sqrt{2} + 2\sqrt{2})\, D\sqrt{n} - 7D$

$= (2\sqrt{2} + 2\sqrt{2})\, D\sqrt{n} - 7D$

$= 4\sqrt{2}\, D\sqrt{n} - 7D$

$= 5.66\, D\, \sqrt{n} - 7D$

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

Fixed-size block CSA (FCSA)

R groups

k bits

$n = R \cdot k$

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.

$(k-1)T_p + T_m + (n/k-2)T_s + (k-1)\ T_p$
$= (6-1)2D + D + (10-2)2D + (4-1)\ 2D$
$= (10 + 1 + 16 + 6)D$
$= 33D$

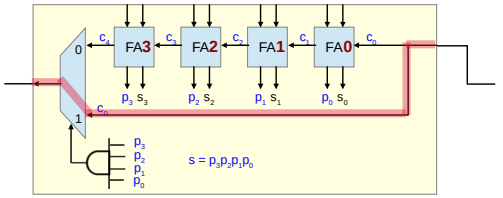
(6-1)2D → (4-1)2D : -4D
37D – 4D = 33D

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

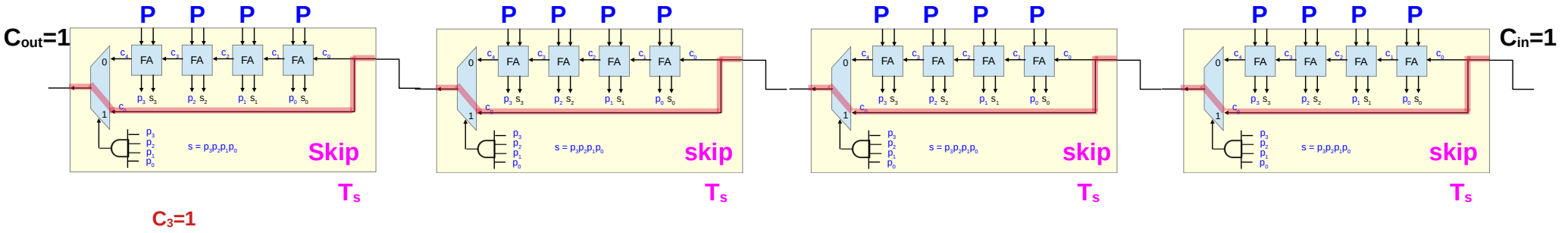
More Fast Adders, Ivor Page, University of Texas at Dallas

Carry Skip Adder

A carry signal centering a certain block
can be propagated past the block
without waiting 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

References

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