Carry Chain Adder (6A)

-
-

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

This document was produced by using OpenOffice and Octave.

Manchester Carry Chain

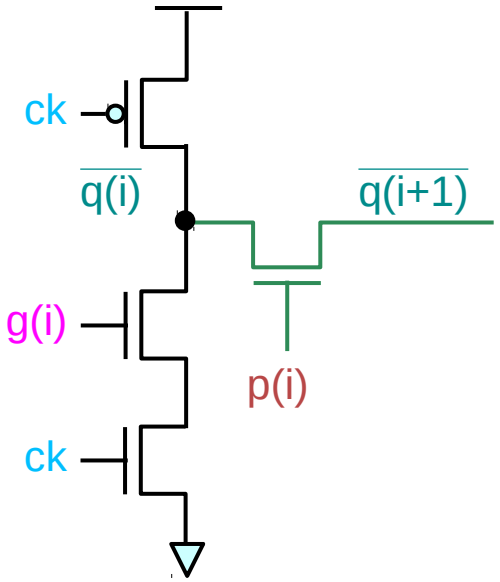
the output node $\overline{q(i+1)}$ is precharged,
when the synchronization signal is equal to 0 (ck=0),

when ck=1, the output node is discharged
if either p(i) =1 and the preceding node $\overline{q(i)}$
has been discharged, or
if g(i)=1.

In order that it works properly
g(i) and p(i) should not be equal to 1 simultaneously
so that the definition of g(i) cannot be relaxed
as in the preceding case

p(i)	0	1
0	0	1
1	1	0

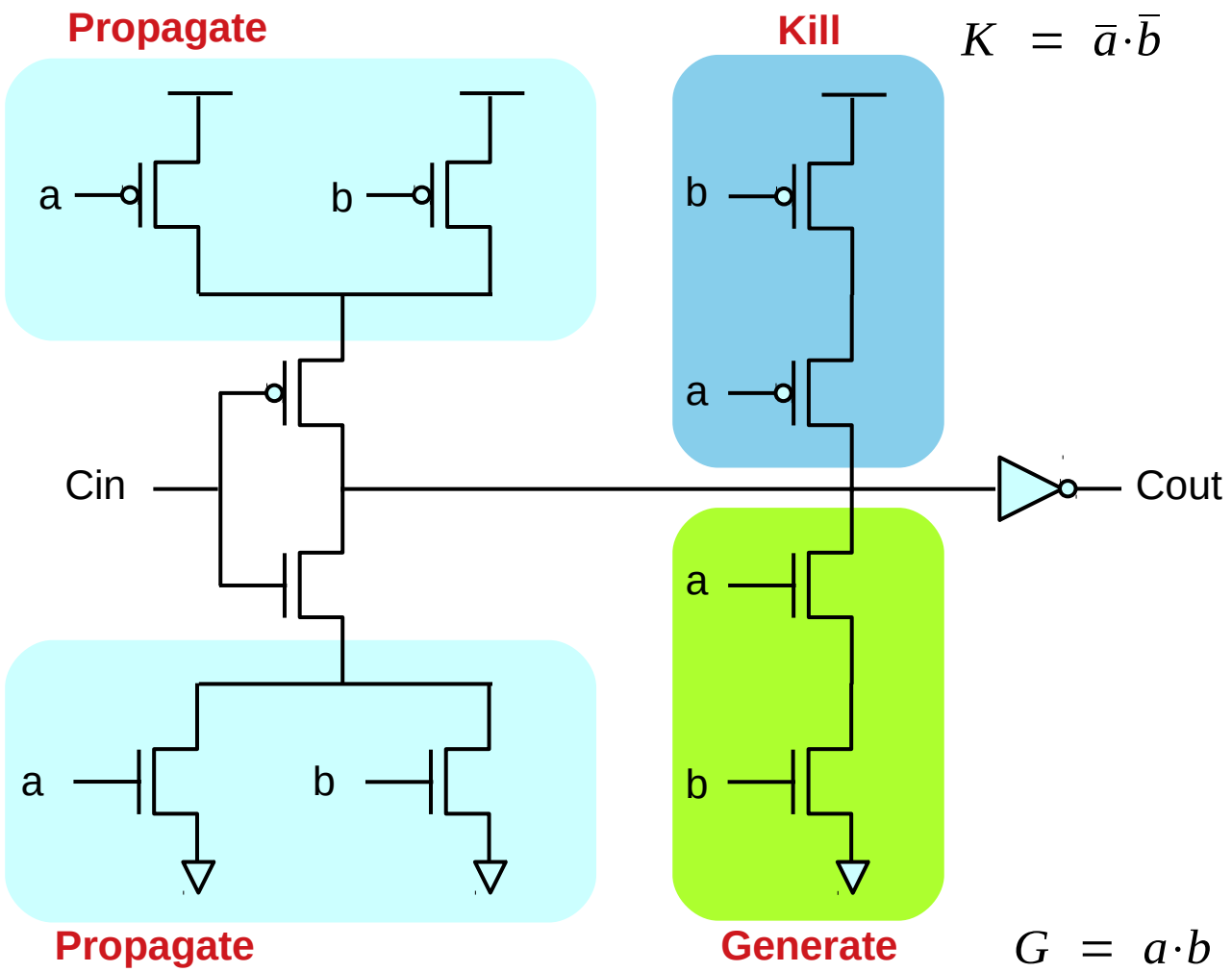
g(i)	0	1
0	0	0
1	0	1



Manchester Carry Chain

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

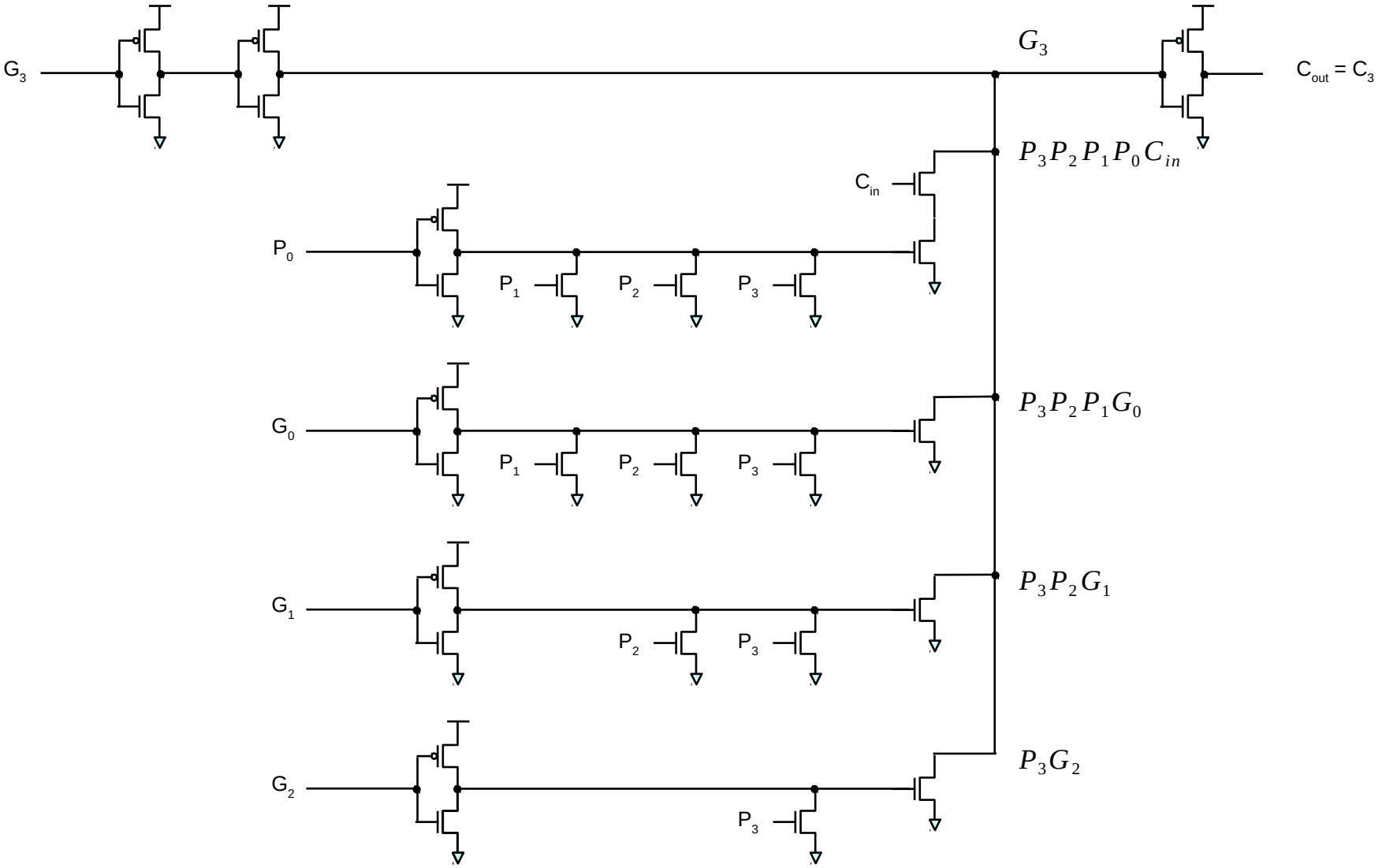
Carry section of FA



Digital Electronics and Design with VHDL,V, A< Pedroni

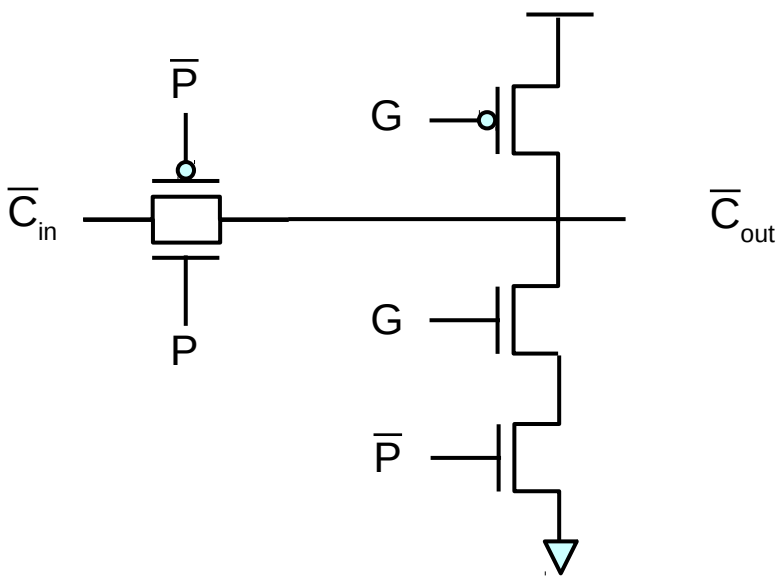
Static Carry Circuit

$$G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 c_{in} = c_3$$



Principles of CMOS VLSI design – A Systems Perspective, N Weste, K Eshraghian

Static Carry Circuit – using G, P



P cannot be relaxed

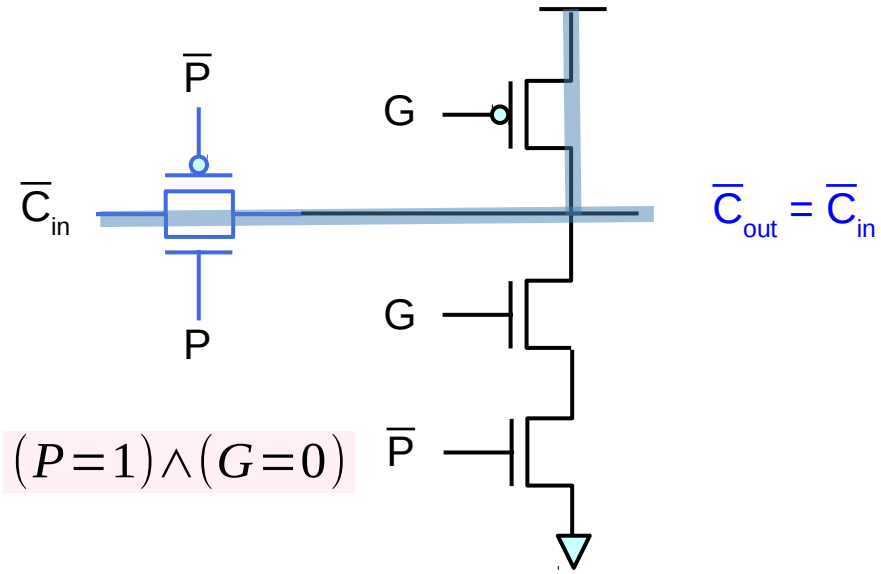
$$P = a \oplus b$$

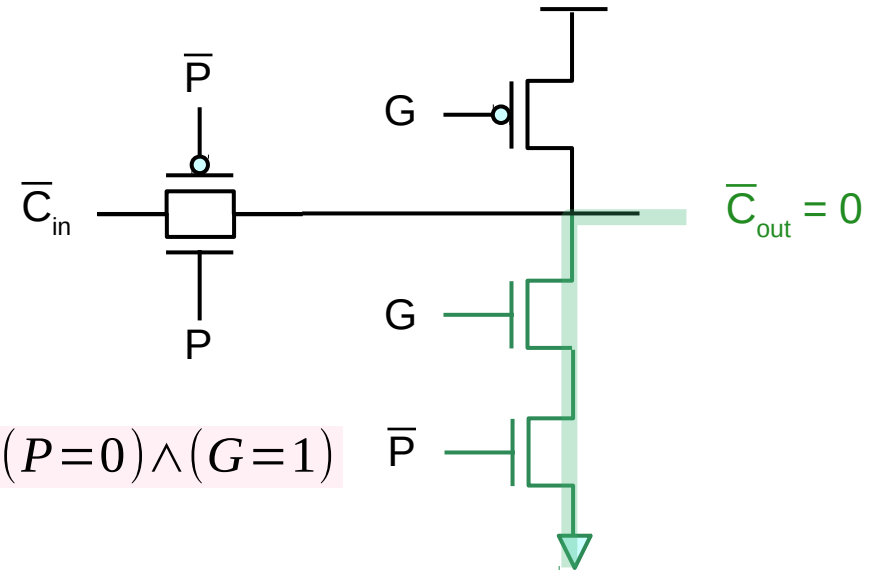
~~$P = a + b$~~

$$G = a \cdot b$$

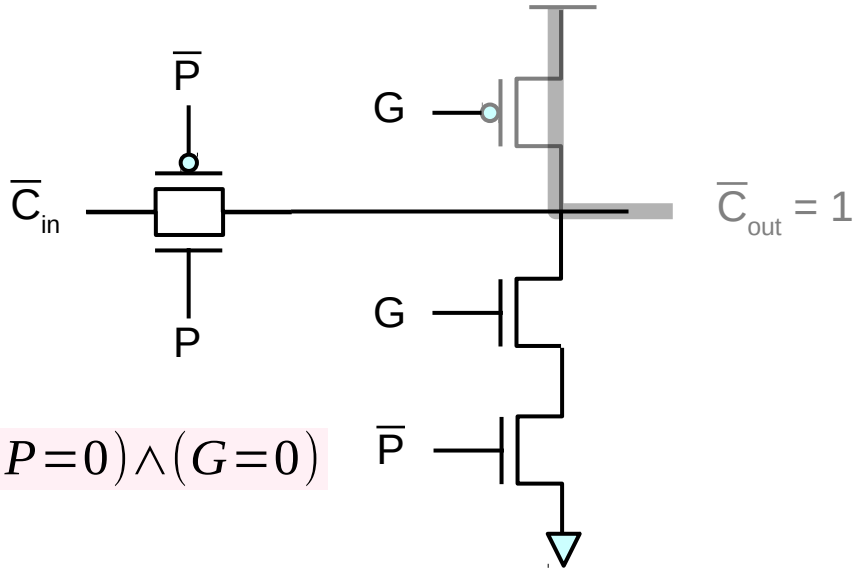
Digital Electronics and Design with VHDL,V, A< Pedroni

Static Carry Circuit – using G, P



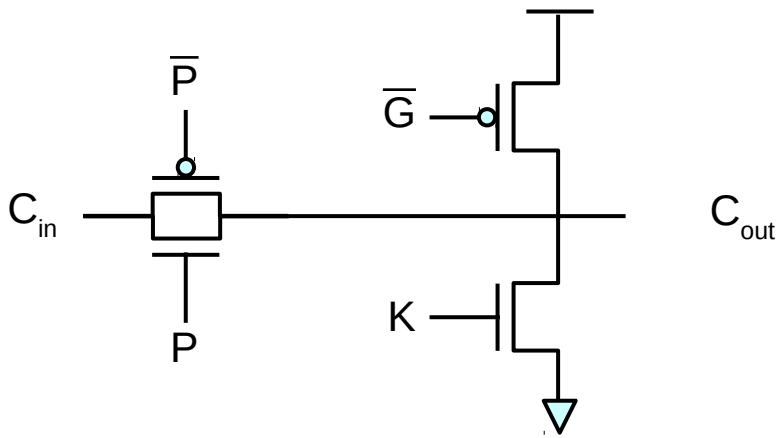


$$P = a \oplus b$$
$$G = a \cdot b$$



Digital Electronics and Design with VHDL,V, A< Pedroni

Static Carry Circuit – using G, P, K



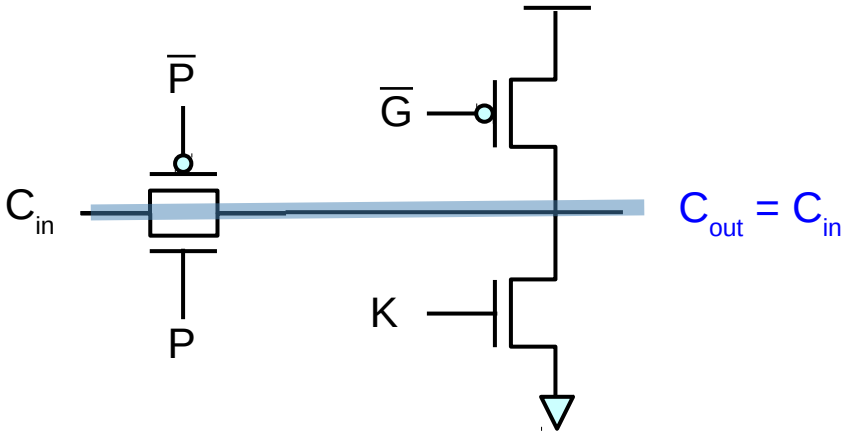
$$P = a \oplus b$$

$$G = a \cdot b$$

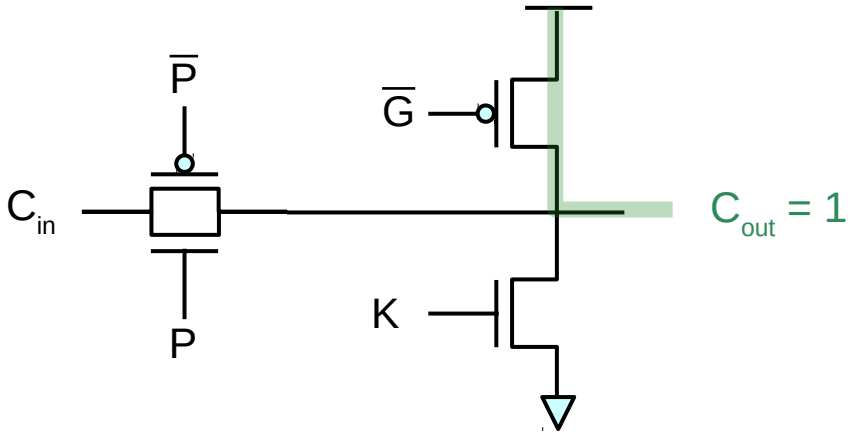
$$K = \bar{a} \cdot \bar{b}$$

Digital Electronics and Design with VHDL,V, A< Pedroni

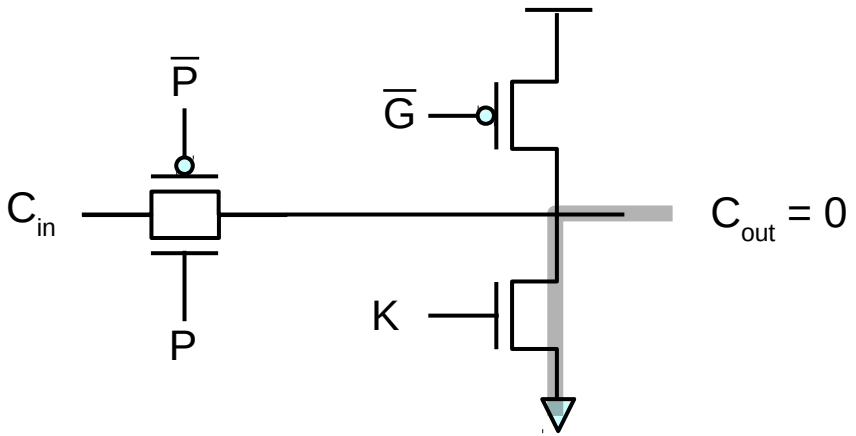
Static Carry Circuit – using G, P, K



$$P = a \oplus b$$



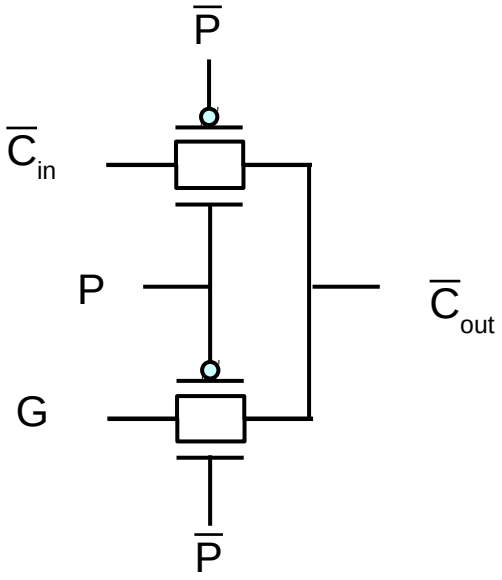
$$G = a \cdot b$$

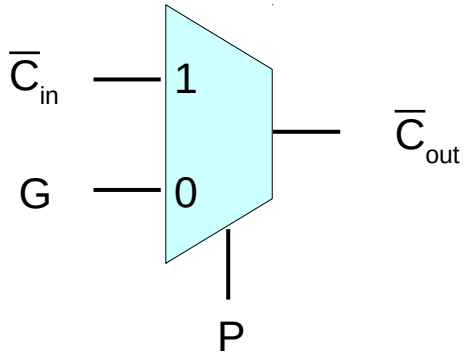


$$K = \bar{a} \cdot \bar{b}$$

Digital Electronics and Design with VHDL,V, A< Pedroni

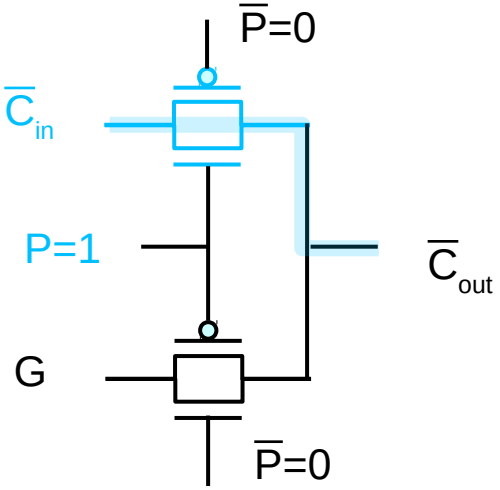
Static Carry Circuit – using Multiplexer

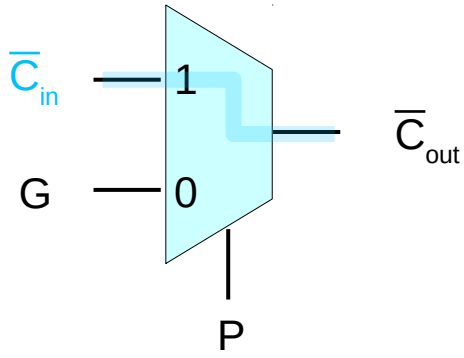


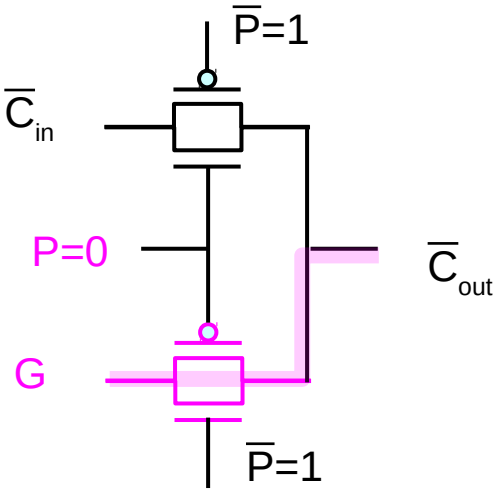


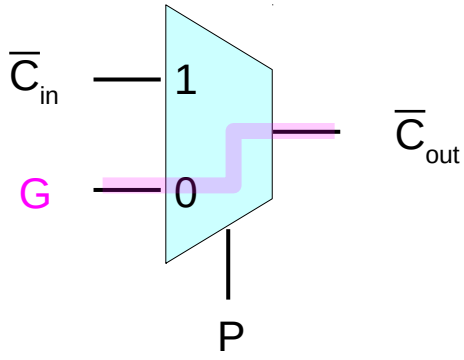
Digital Electronics and Design with VHDL,V, A< Pedroni

Static Carry Circuit – using Multiplexer



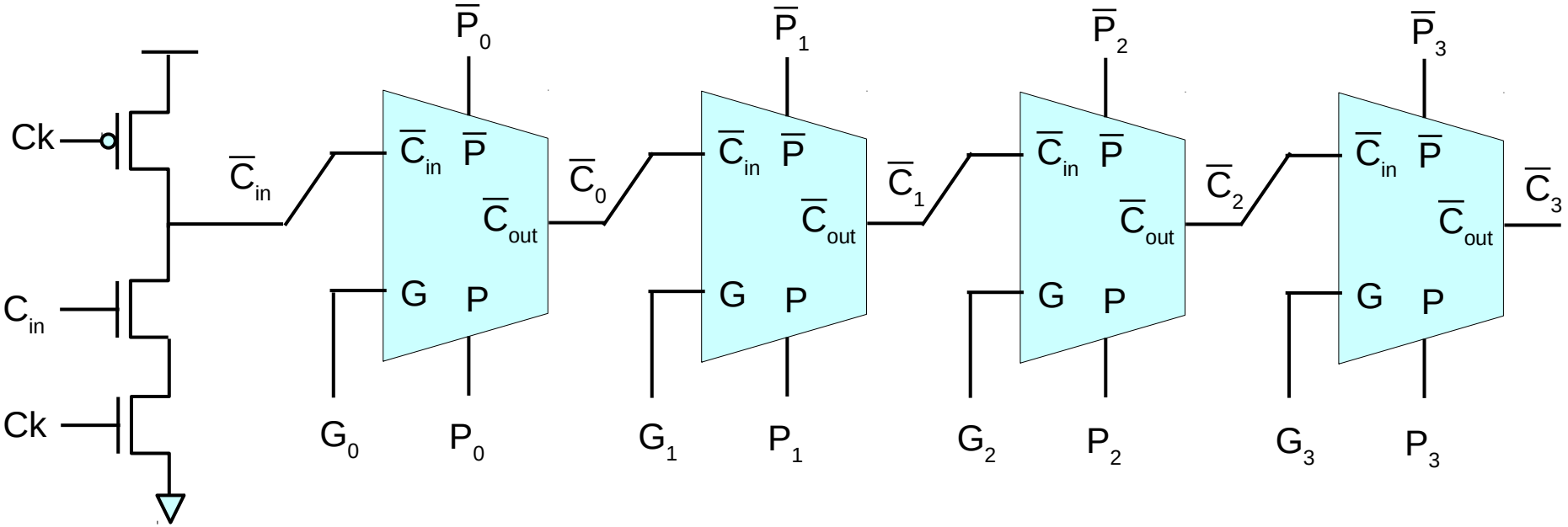


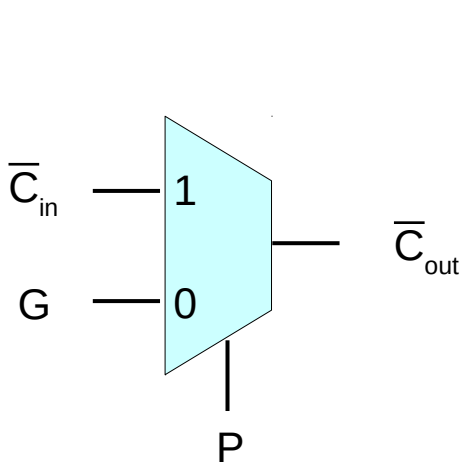


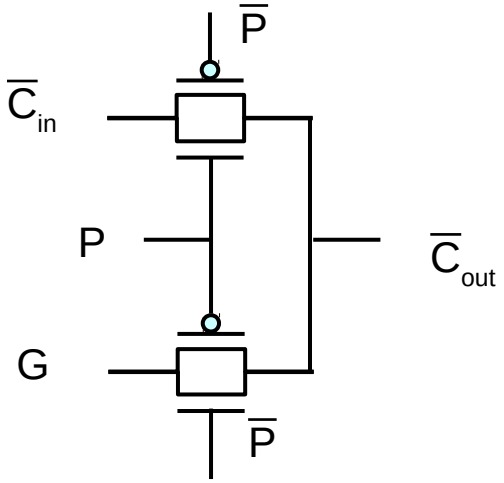


Digital Electronics and Design with VHDL,V, A< Pedroni

Static Carry Circuit – using multiplexers

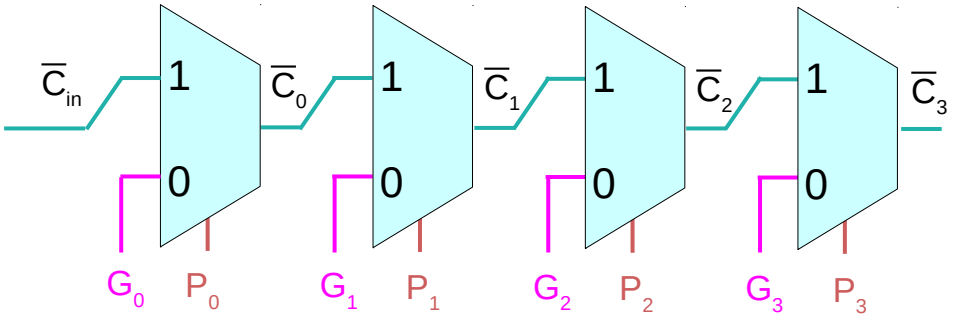


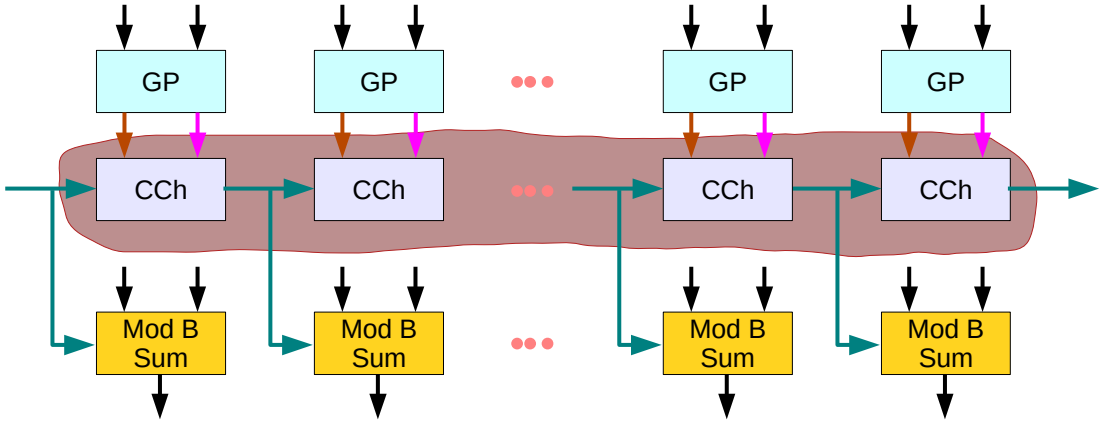




Digital Electronics and Design with VHDL,V, A< Pedroni

Static Carry Circuit – using multiplexers





Principles of CMOS VLSI design – A Systems Perspective, N Weste, K Eshraghian

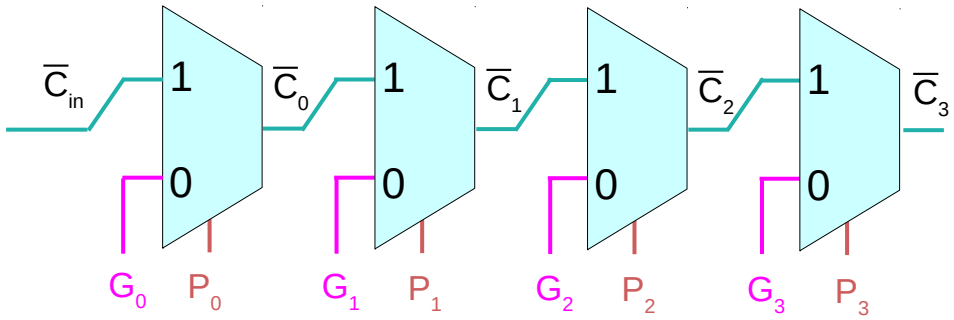
Static Carry Circuit – using multiplexers

A multiplexer-based 4-bit adder

cascading four such stages
supplying P_i, G_i

This is commonly called
a **Manchester carry adder**

not **Manchester carry chain adder**

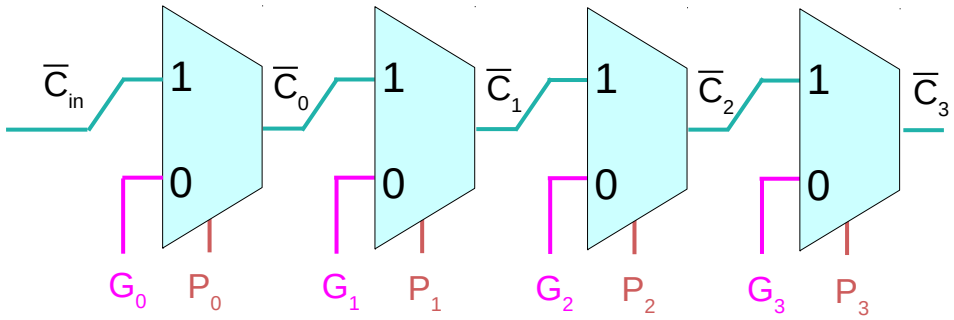


There is some similarity with the domino carry circuit (nMos)
Manchester carry chain adder

However, the intermediate carry gates are no longer needed, (G_i, P_i)
Because the carry values are available in a distributed fashion

Principles of CMOS VLSI design – A Systems Perspective, N Weste, K Eshraghian

Static Carry Circuit – using multiplexers



The 4-bit adder is chosen to reduce
the number of series-propagate transistors
Which improves the speed

Note that if all propagate signals are true,
and CI is high , six series n-transistors
Pull the output node lo in the case of the dynamic gate
While five transistors re in series in the static gate

Principles of CMOS VLSI design – A Systems Perspective, N Weste, K Eshraghian

Manchester Carry Chain – Dynamic Logic

However, not all logic families have these **internal nodes**,
CMOS being a major example.

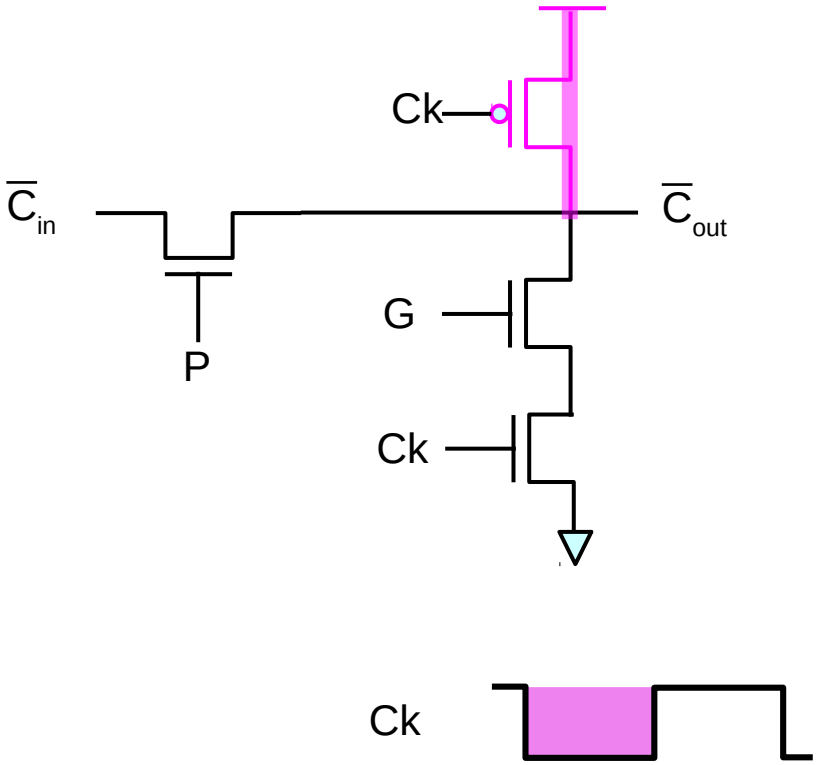
Dynamic logic can support shared logic,
as can transmission gate logic.

One of the major downsides of the Manchester carry chain
is that the **capacitive load** of all of these outputs,
together with the resistance of the transistors causes
the **propagation delay** to increase much *more quickly*
than a regular carry lookahead.

A Manchester-carry-chain section generally
doesn't exceed 4 bits.

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

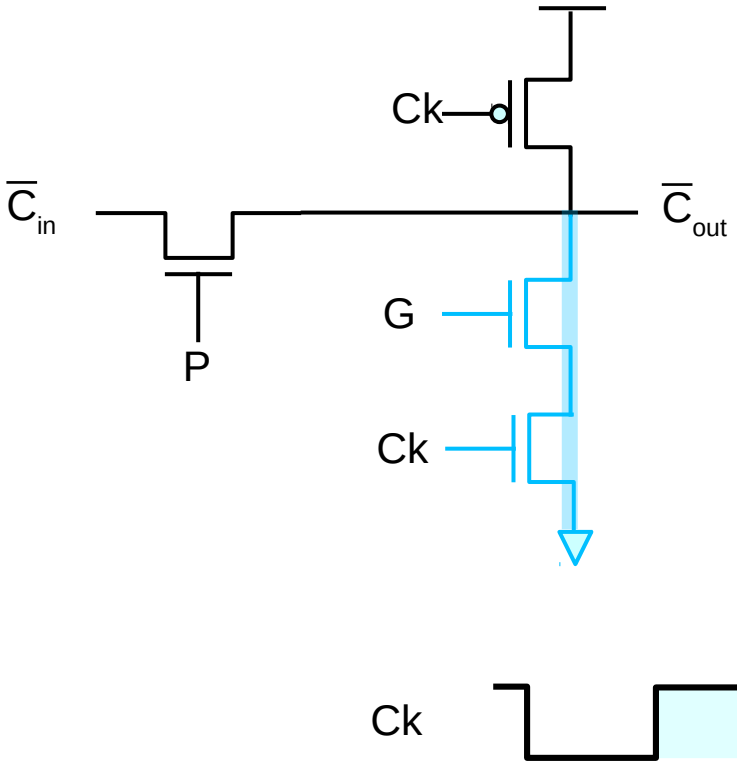
Dynamic Carry Circuit – using G, P



When **CLK** is **low**, the output node
is precharged by the **pull-up** transistor

Principles of CMOS VLSI design – A Systems Perspective, N Weste, K Eshraghian

Dynamic Carry Circuit – using G, P

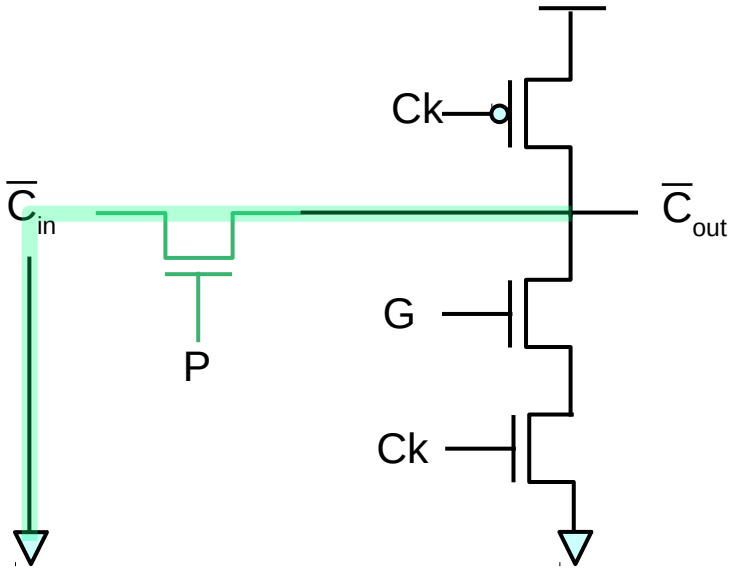


When **CLK** goes **high**,
the n **pull-down** transistor turns on
If carry generate **G=AB** is **true**,
then the output node **discharge**

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

Principles of CMOS VLSI design – A Systems Perspective, N Weste, K Eshraghian

Dynamic Carry Circuit – using G, P



When **CLK** goes **high**,
the n **pull-down** transistor turns on
If carry propagate **P=A+B** is **true**,
then a **previous carry** may be coupled
to the output node,
conditionally **discharging** it

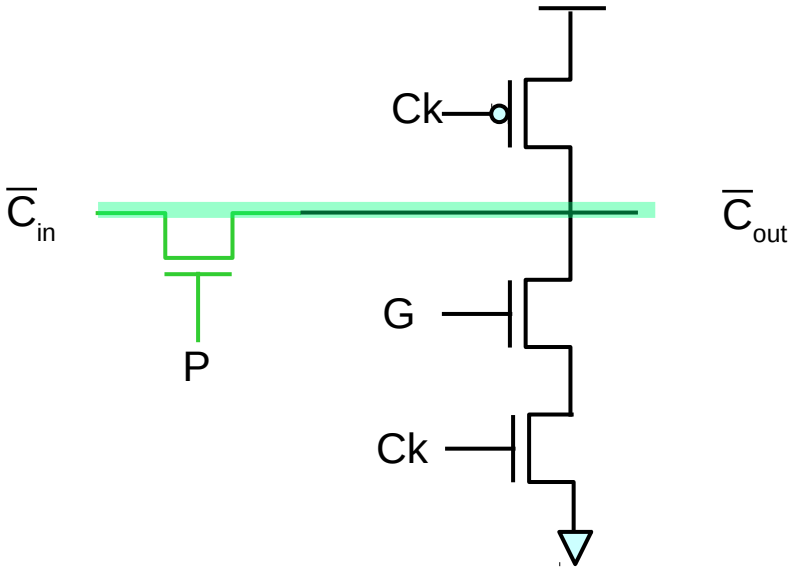
Note that in this circuit CARRY is actually propagated

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

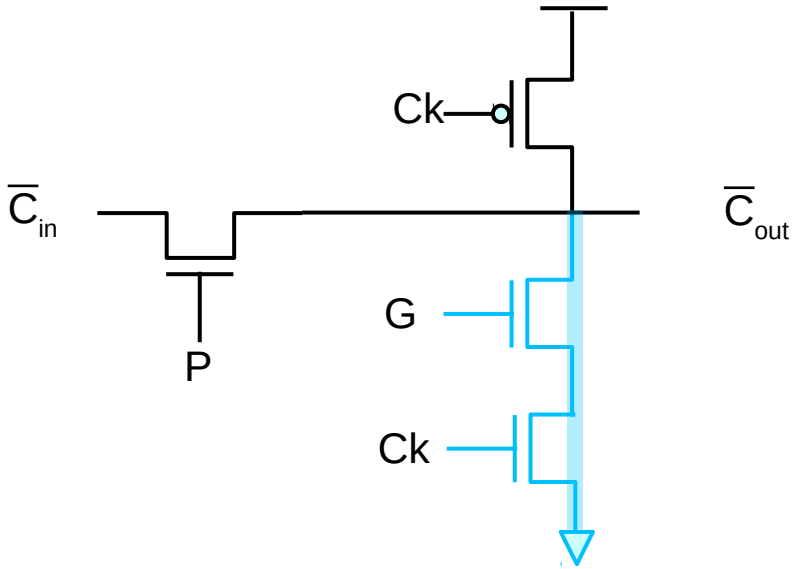
Principles of CMOS VLSI design – A Systems Perspective, N Weste, K Eshraghian

Dynamic Carry Circuit – using G, P

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



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



This requires P must not be relaxed

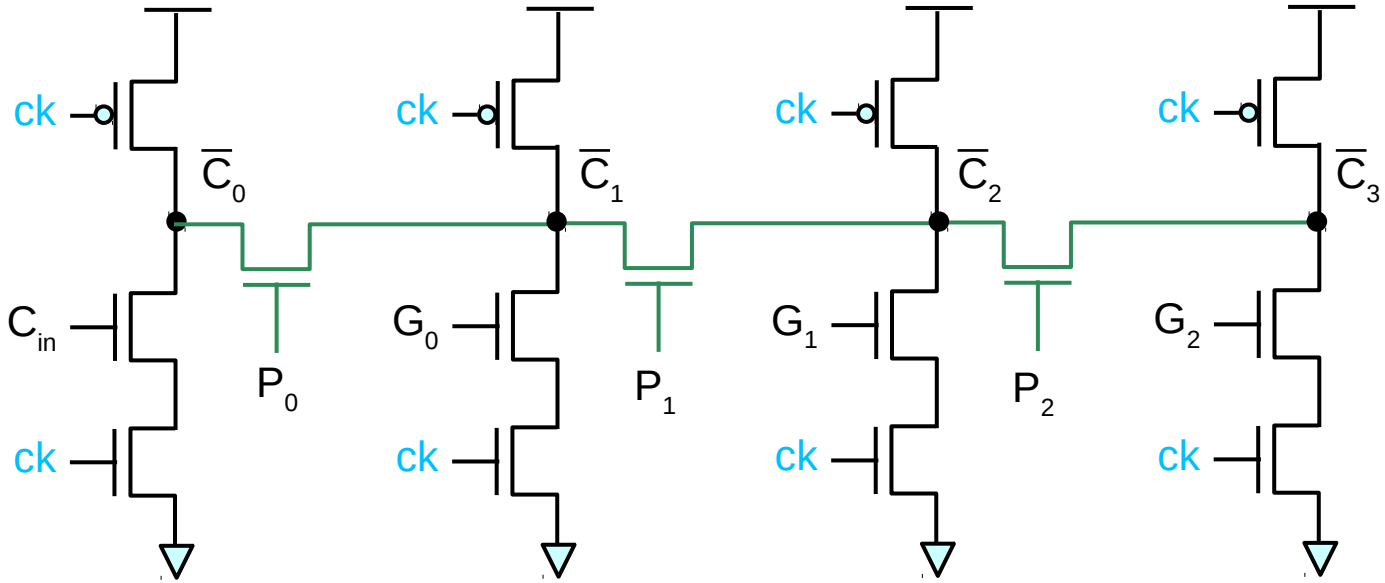
$$P = a \oplus b$$

p(i)	0	1
0	0	1
1	1	0

g(i)	0	1
0	0	0
1	0	1

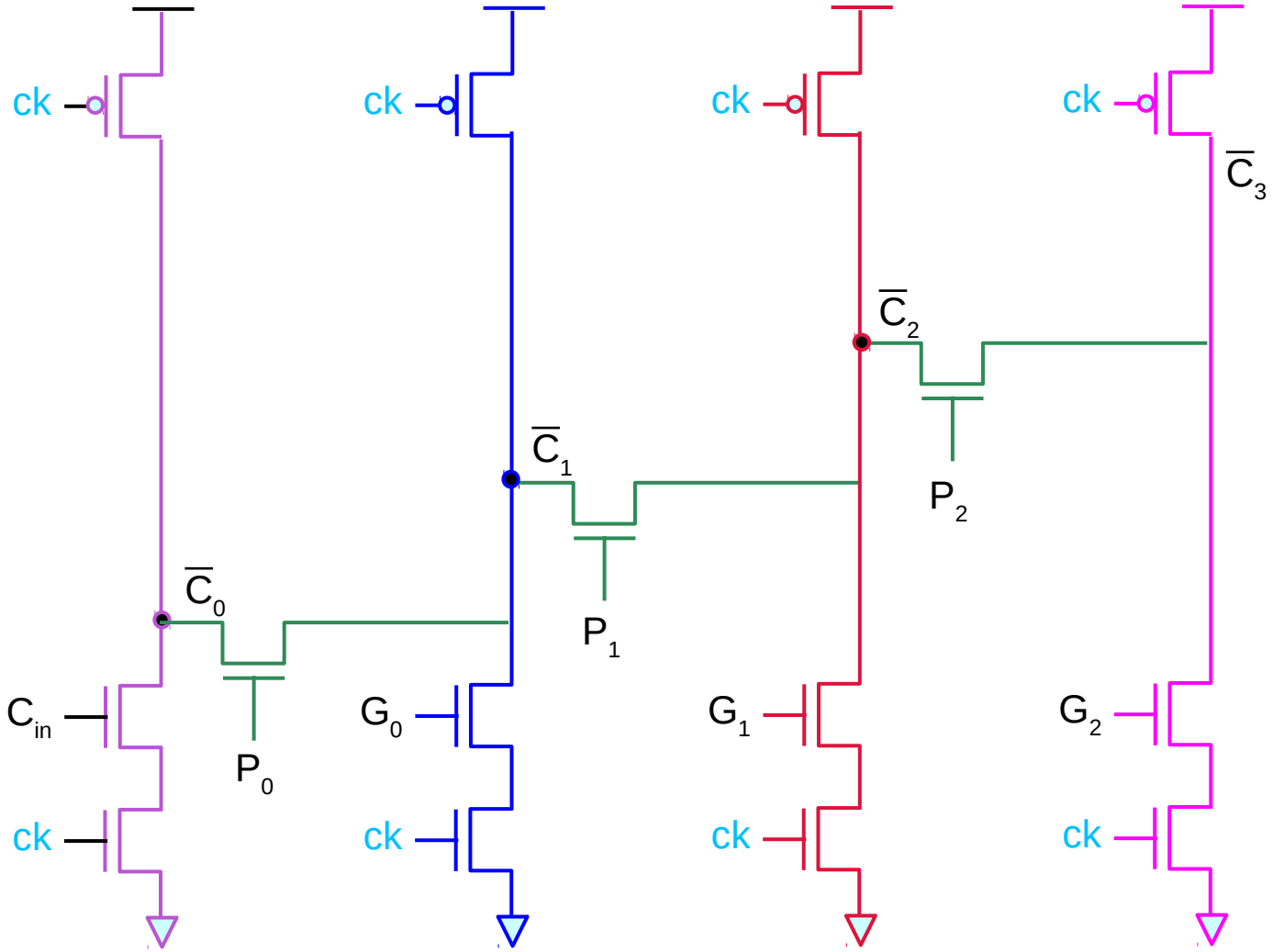
Digital Electronics and Design with VHDL,V, A< Pedroni

Dynamic Manchester Carry-Chain Adder



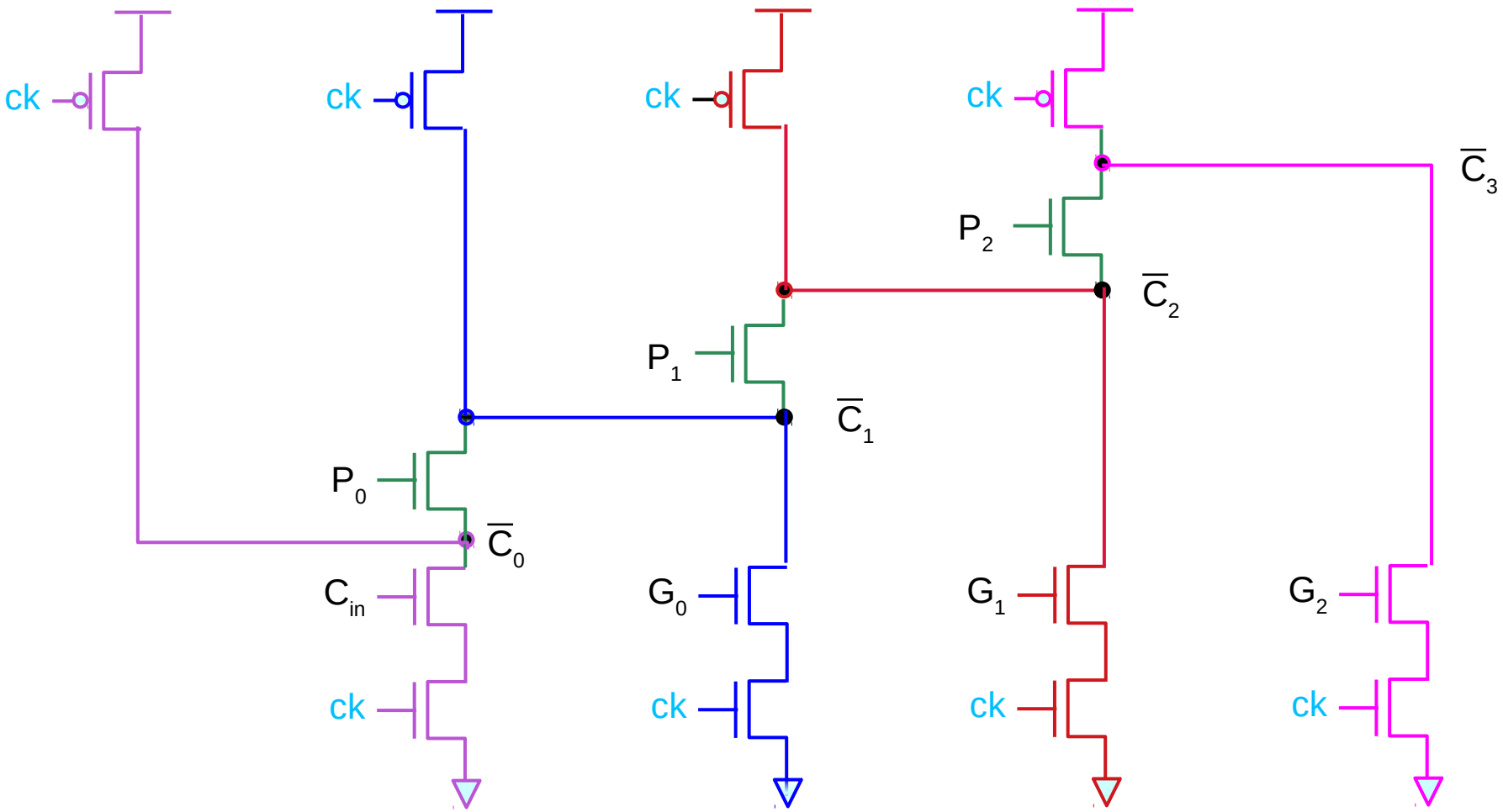
Digital Electronics and Design with VHDL,V, A< Pedroni

Other representation I



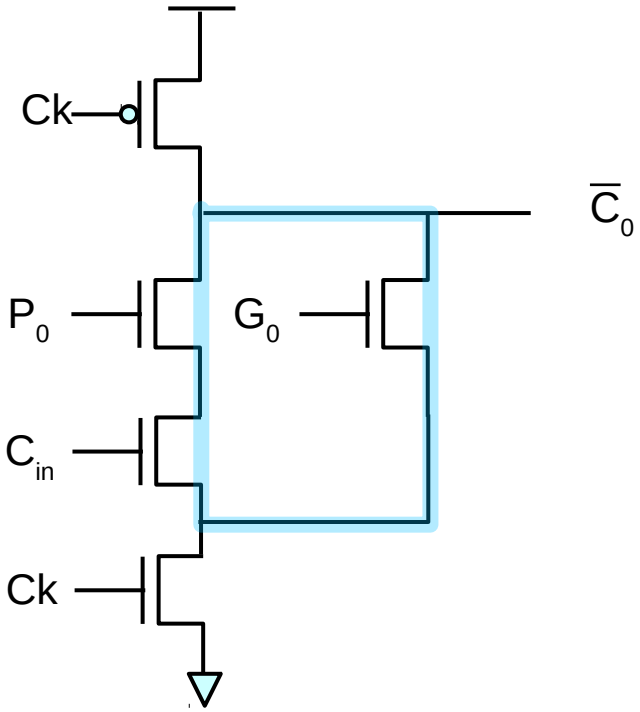
Digital Electronics and Design with VHDL,V, A< Pedroni

Other representation II



Digital Electronics and Design with VHDL,V, A< Pedroni

Dynamic Carry Circuit – C_0

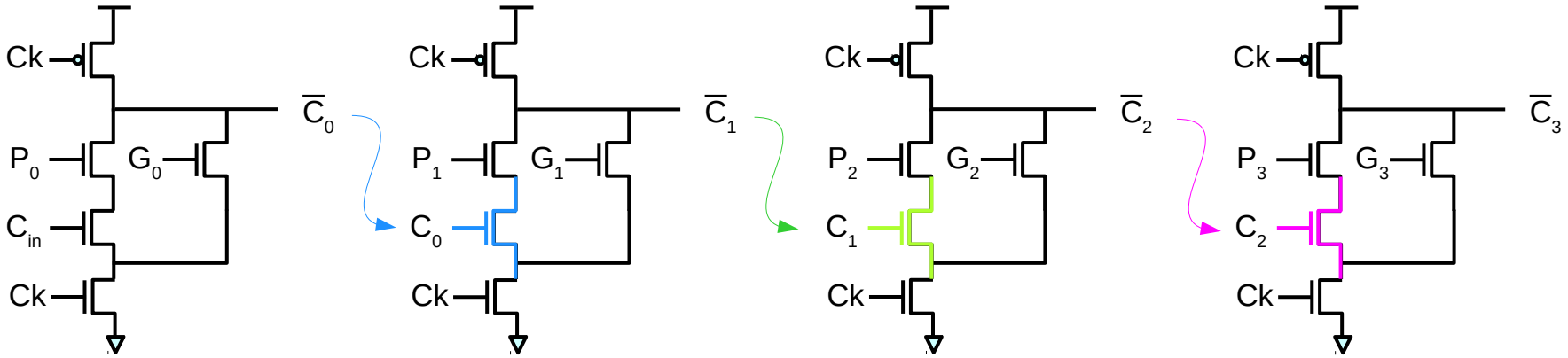


$$c_0 = G_0 + P_0 c_{in}$$

$$c_{out} = P c_i + G$$

Principles of CMOS VLSI design – A Systems Perspective, N Weste, K Eshraghian

Dynamic Carry Circuit – C_0, C_1, C_2, C_3



$$c_0 = G_0 + P_0 c_{in}$$

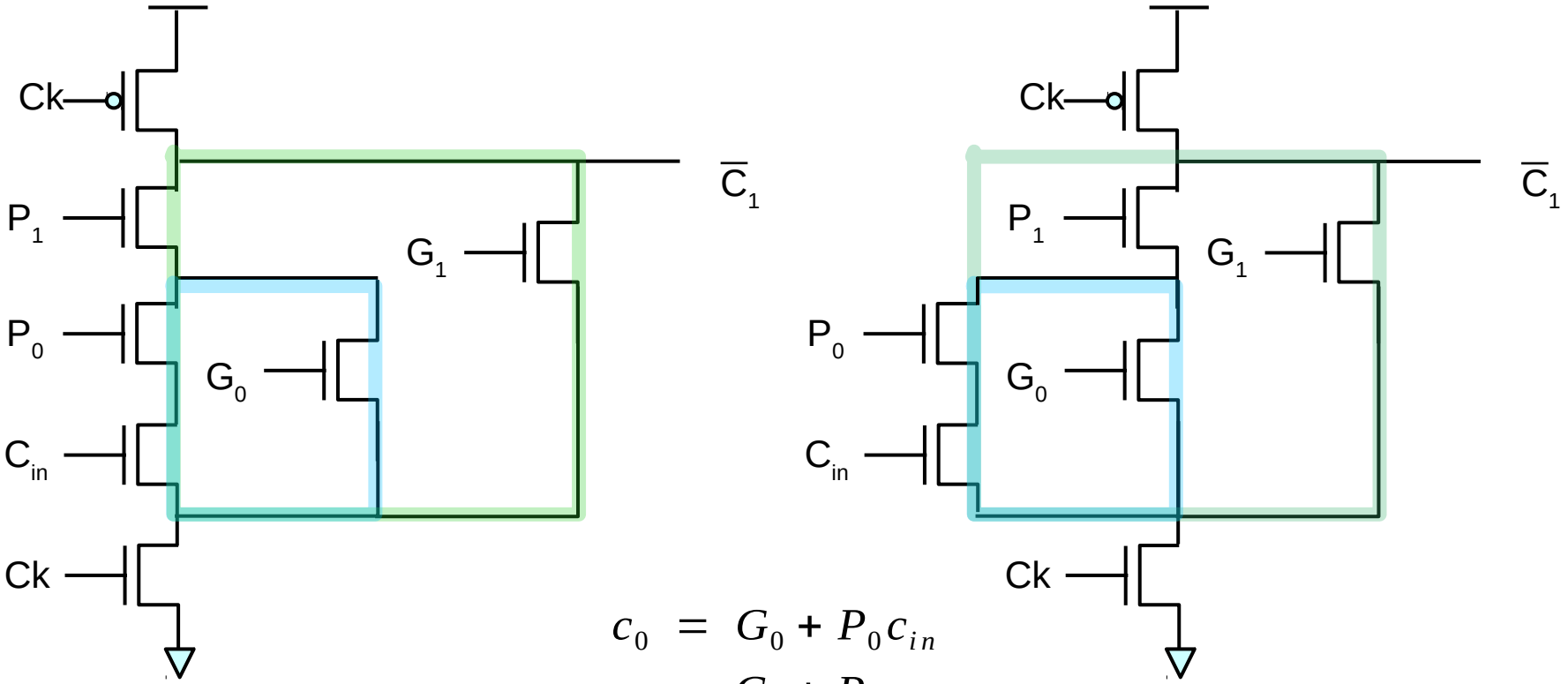
$$c_1 = G_1 + P_1 c_0$$

$$c_2 = G_2 + P_2 c_1$$

$$c_3 = G_3 + P_3 c_2$$

Principles of CMOS VLSI design – A Systems Perspective, N Weste, K Eshraghian

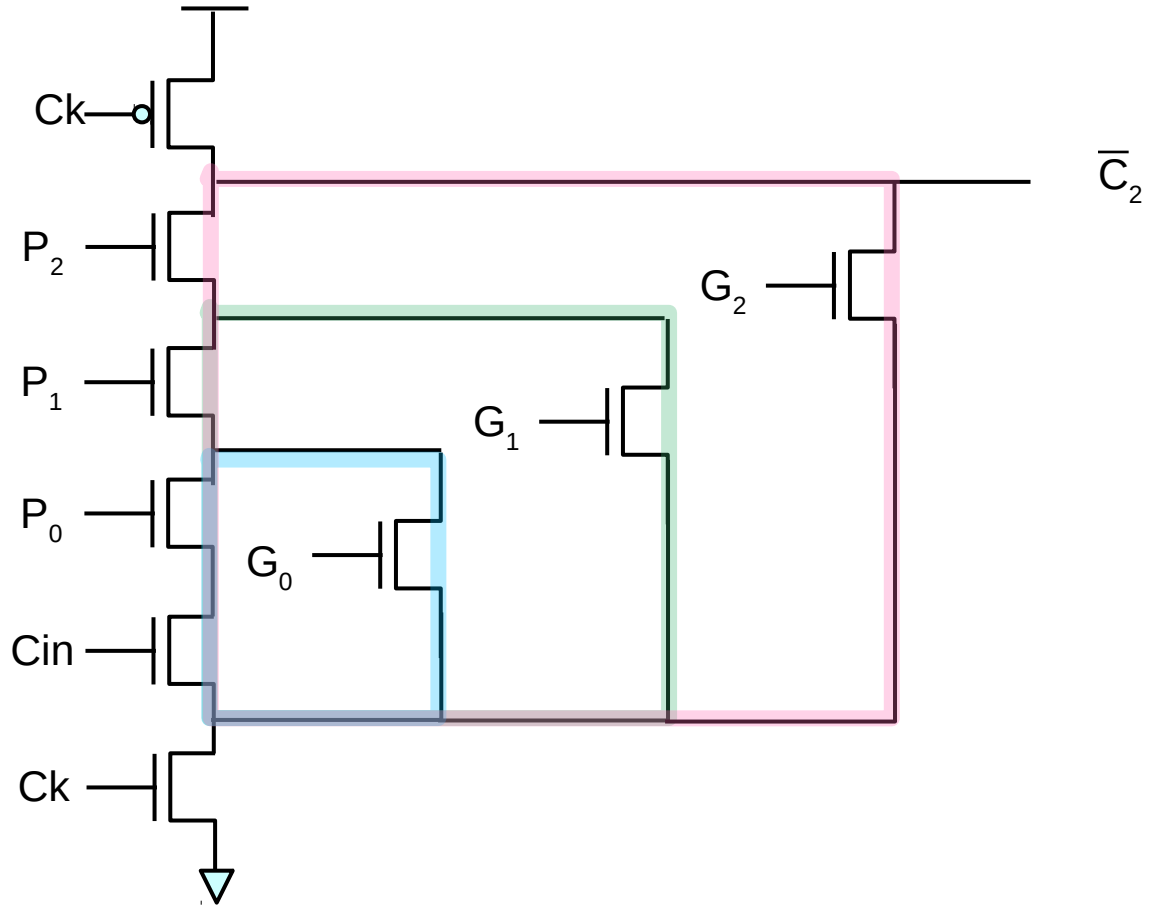
Dynamic Carry Circuit – C_1



$$c_0 = G_0 + P_0 c_{in}$$
$$c_1 = G_1 + P_1 c_0$$

Principles of CMOS VLSI design – A Systems Perspective, N Weste, K Eshraghian

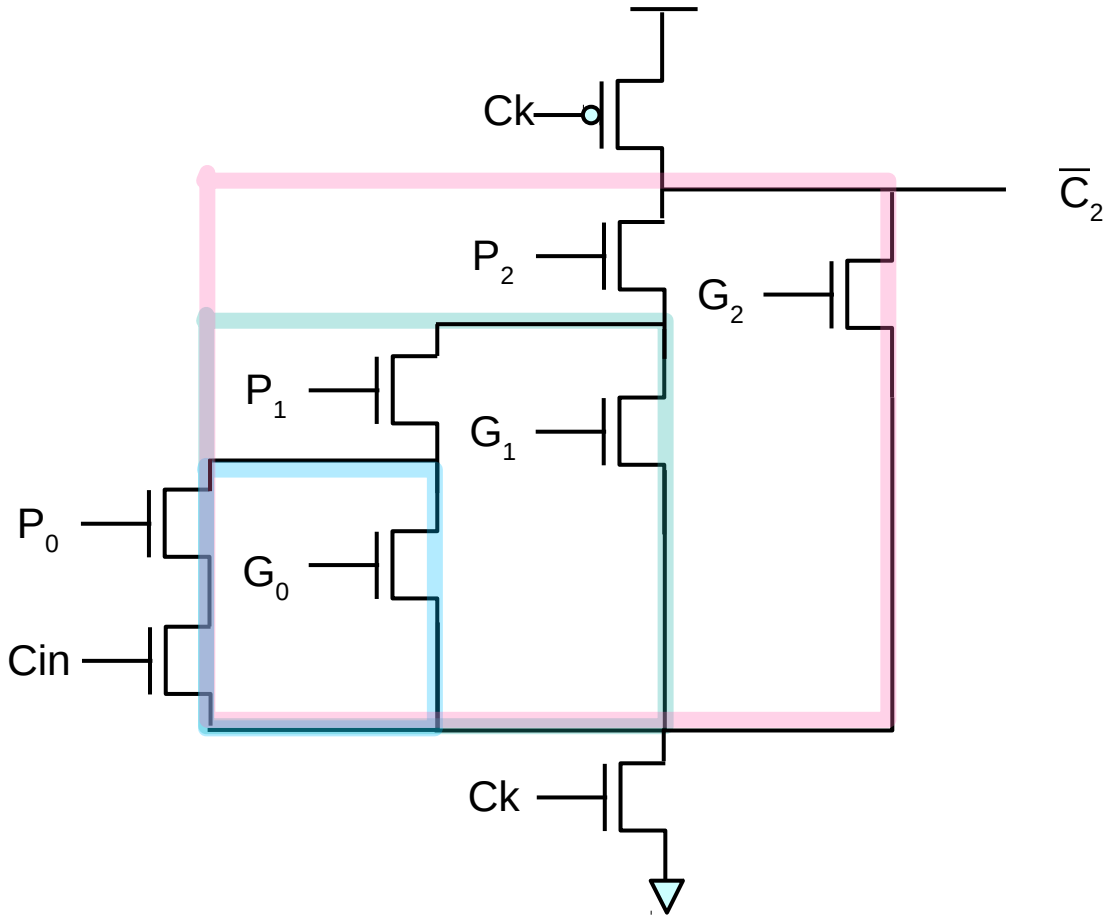
Dynamic Carry Circuit – C_2 (1)



$$c_0 = G_0 + P_0 c_{in}$$
$$c_1 = G_1 + P_1 c_0$$
$$c_2 = G_2 + P_2 c_1$$

Principles of CMOS VLSI design – A Systems Perspective, N Weste, K Eshraghian

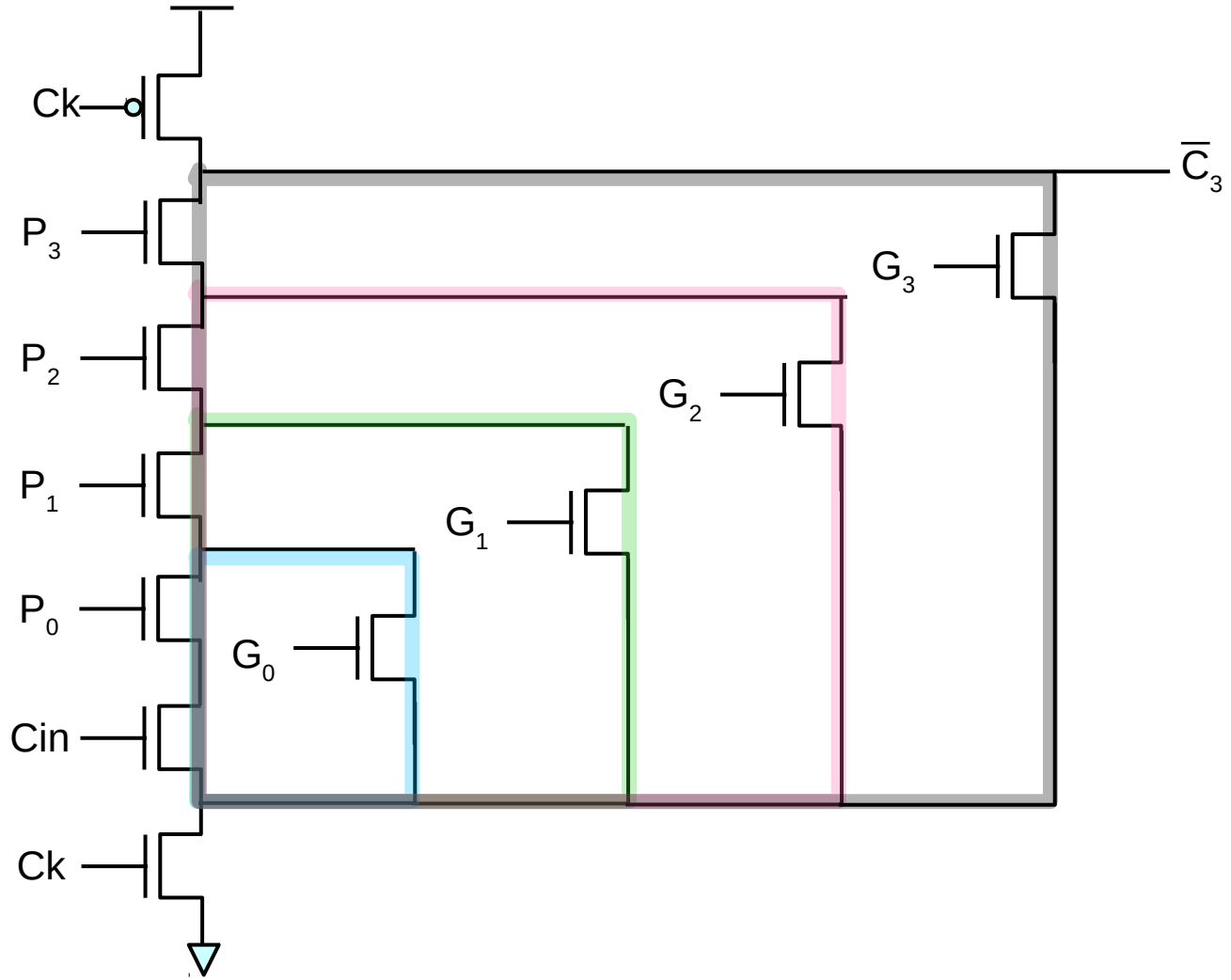
Dynamic Carry Circuit – C_2 (2)



$$c_0 = G_0 + P_0 c_{in}$$
$$c_1 = G_1 + P_1 c_0$$
$$c_2 = G_2 + P_2 c_1$$

Principles of CMOS VLSI design – A Systems Perspective, N Weste, K Eshraghian

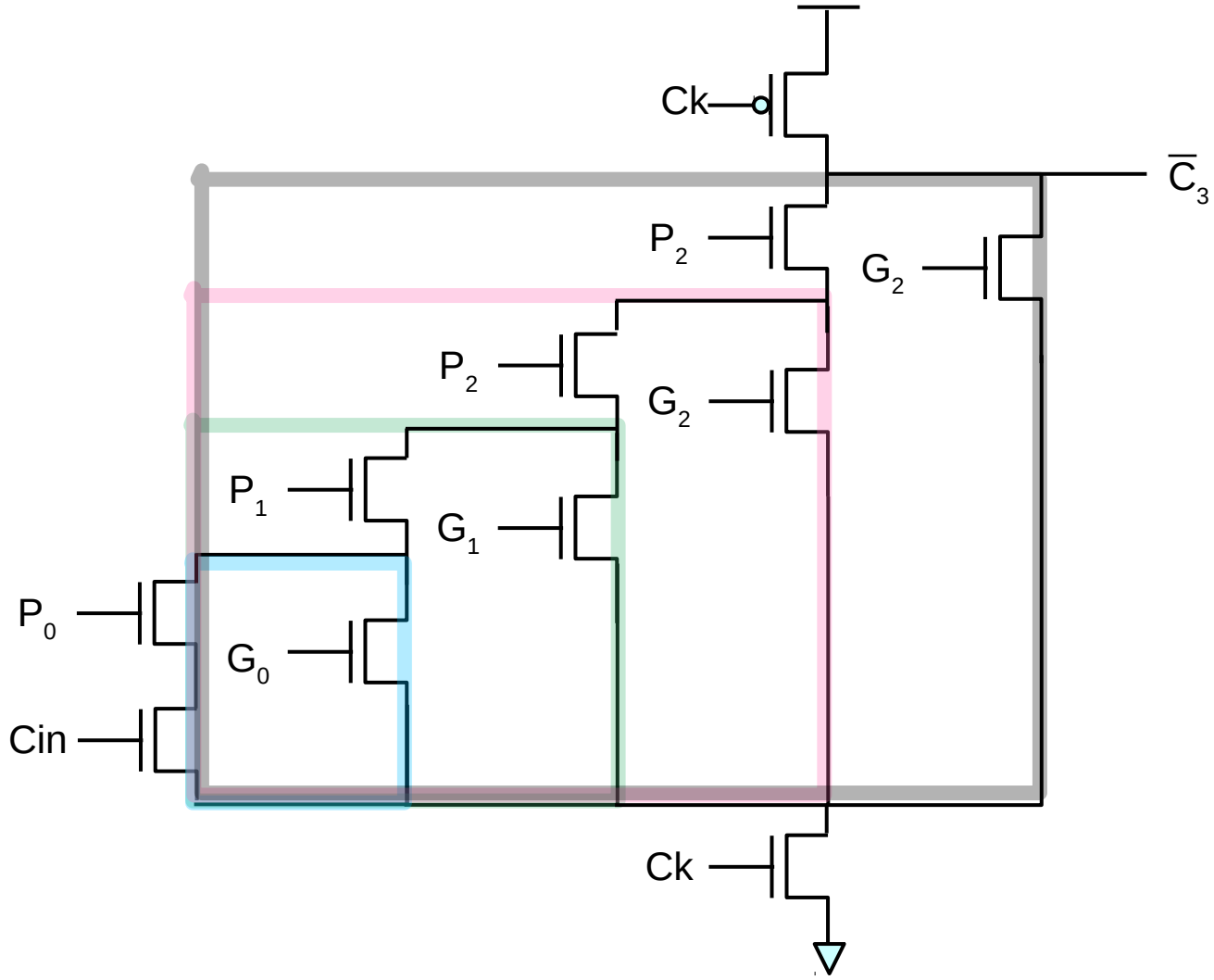
Dynamic Carry Circuit – C_3 (1)



$$c_0 = G_0 + P_0 c_{in}$$
$$c_1 = G_1 + P_1 c_0$$
$$c_2 = G_2 + P_2 c_1$$
$$c_3 = G_3 + P_3 c_2$$

Principles of CMOS VLSI design – A Systems Perspective, N Weste, K Eshraghian

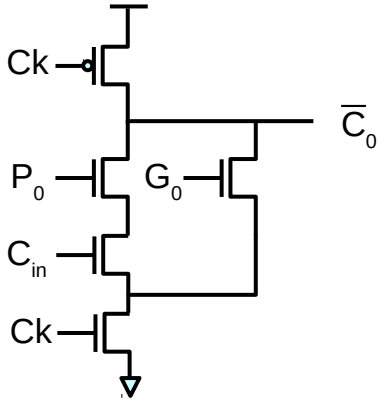
Dynamic Carry Circuit – C_3 (2)



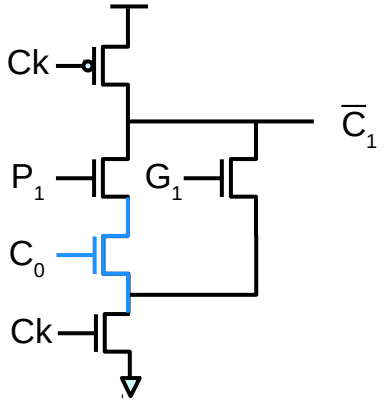
$$c_0 = G_0 + P_0 c_{in}$$
$$c_1 = G_1 + P_1 c_0$$
$$c_2 = G_2 + P_2 c_1$$
$$c_3 = G_3 + P_3 c_2$$

Principles of CMOS VLSI design – A Systems Perspective, N Weste, K Eshraghian

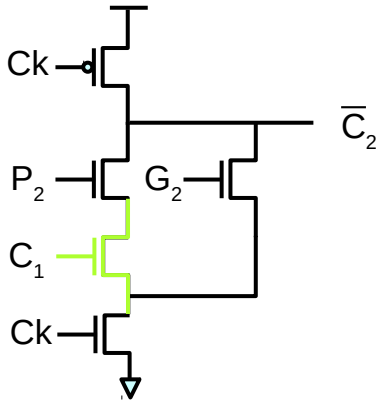
Dynamic Carry Circuit – C_3 (3)



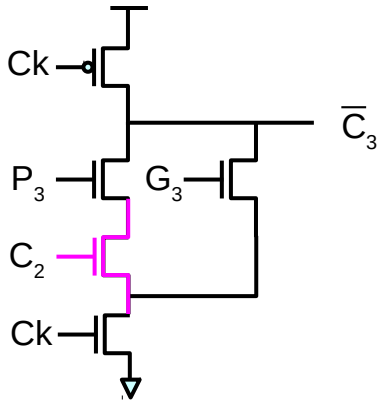
$$c_0 = G_0 + P_0 c_{in}$$



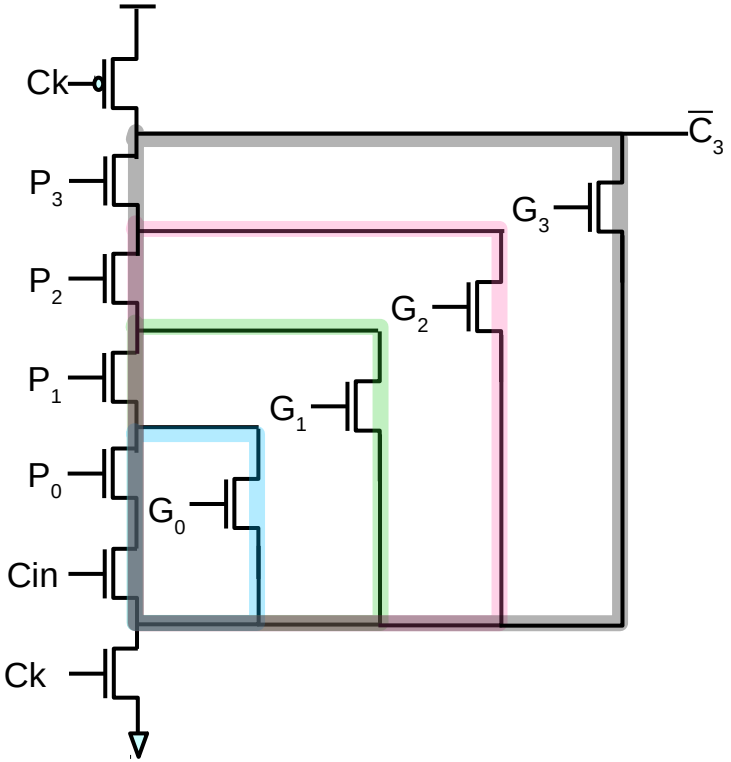
$$c_1 = G_1 + P_1 c_0$$



$$c_2 = G_2 + P_2 c_1$$



$$c_3 = G_3 + P_3 c_2$$



Principles of CMOS VLSI design – A Systems Perspective, N Weste, K Eshraghian

References

[1] http://en.wikipedia.org/
[2] J-P Deschamps,et. al., “Sunthesis of Arithmetic Circuits”, 2006