AC-CS 1.4. Parallel Computation of Sum

1.4.1. Multilevel Carry Lookahead Adder

Full Carry Lookahead Adder (F-CLA)

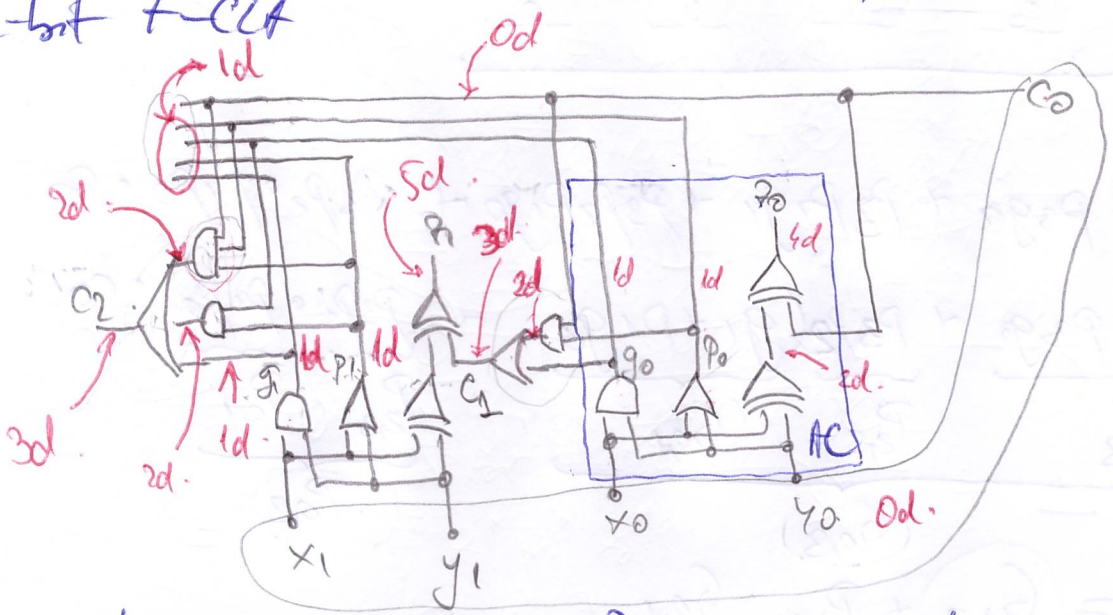
$c_{i+1} = x_i \cdot y_i + c_i(x_i + y_i)$ ⟷ $g_i = x_i \cdot y_i$ generate; $p_i = x_i + y_i$ propagate

$c_{i+1} = g_i + p_i c_i =$ $s_i = x_i \oplus y_i \oplus c_i$

$g_i + p_i g_{i-1} + p_i p_{i-1} c_{i-1} = \dots$

$g_i + p_i g_{i-1} + p_i p_{i-1} g_{i-2} + \dots + p_i p_{i-1} p_{i-2} \dots p_1 g_0 + p_i p_{i-1} p_{i-2} \dots p_1 p_0 c_0$

2-bit F-CLA



$c_1 = g_0 + p_0 c_0$

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

$D^{cout}_{F\text{-}CLA\text{-}n} = 3d$ $\quad D^{s}_{F\text{-}CLA\text{-}n} = 5d$

! Disadvantages: ⇒ fan-out ↑, fan-in ↑

c_0 ?? 2-bit – 3 parts

2-bit: ⇒ OR – 3 inputs, AND – 3 inputs

- only for small n

for $n = 8$ → OR: 9 inputs

⇒ ML-CLA.

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

g_i: carry is generated in rank i

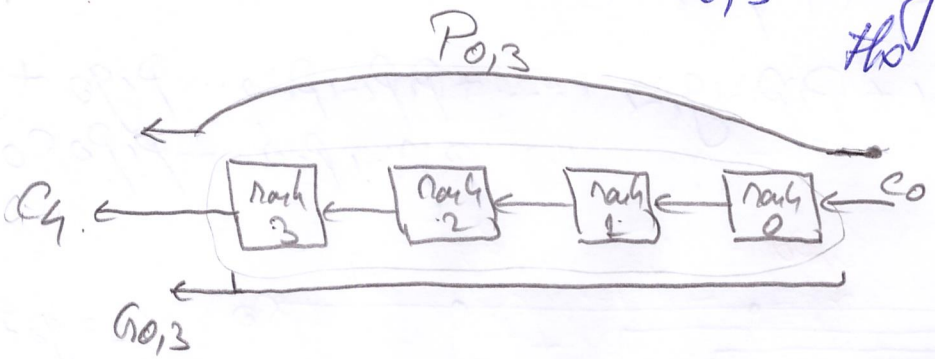
p_i: carry in is propagated to c_{i+1}

$$c_4 = g_3 + p_3 c_3 =$$
$$g_3 + p_3 g_2 + p_3 p_2 c_2 = \dots$$
$$\underbrace{g_3 + p_3 g_2 + p_3 p_2 g_1 + p_3 p_2 p_1 g_0}_{G_{0,3}} + \underbrace{p_3 p_2 p_1 p_0}_{P_{0,3}} \cdot c_0$$

$$c_4 = G_{0,3} + P_{0,3} \cdot c_0$$

$G_{0,3}$: carry generated in block of ranks from 0 to 3

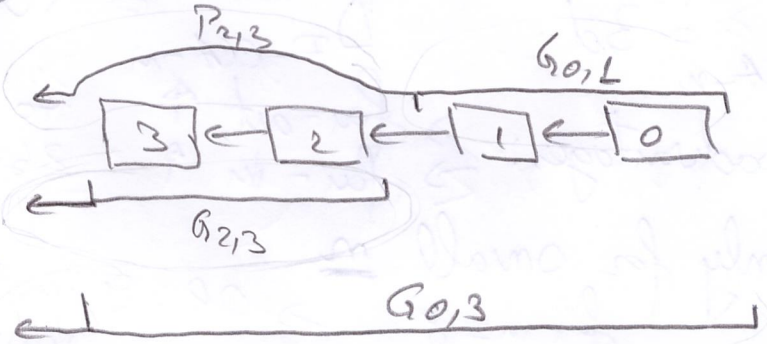
$P_{0,3}$: carry was propagated over the block of ranks 0–3



$$c_4 = g_3 + p_3 g_2 + p_3 p_2 g_1 + p_3 p_2 p_1 g_0 + p_3 p_2 \cdot p_1 p_0 \cdot c_0$$
$$= \underbrace{g_3 + p_3 g_2}_{G_{2,3}} + \underbrace{p_3 p_2}_{P_{2,3}} \underbrace{(g_1 + p_1 g_0)}_{G_{0,1}} + \underbrace{p_3 p_2}_{P_{2,3}} \cdot \underbrace{p_1 p_0}_{P_{0,1}} \cdot c_0$$

$G_{0,3}$; $P_{0,3}$

$$G_{0,3} = G_{2,3} + P_{2,3} \cdot G_{0,1}$$



$$P_{0,3} = P_{2,3} \cdot P_{0,1}$$

Notation: $G_{i,i} = g_i = x_i \cdot y_i$

$P_{i,i} = p_i = x_i + y_i$

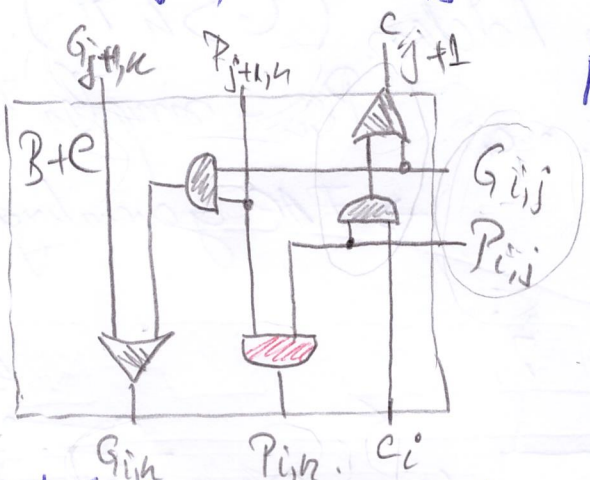
In general:

AC: $C_{k+1} = G_{i,k} + P_{i,k} \cdot C_i$, $i \le k$

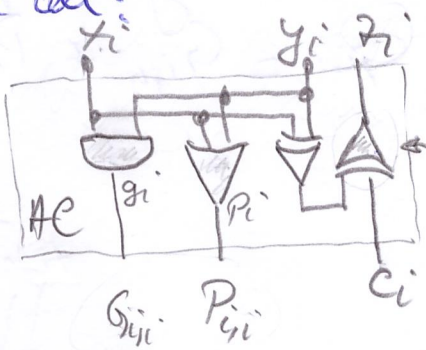
B+C: $G_{i,k} = G_{j+1,k} + P_{j+1,k} \cdot G_{i,j}$, $i \le j < k$

$P_{i,k} = P_{j+1,k} \cdot P_{i,j}$, $i \le j < k$

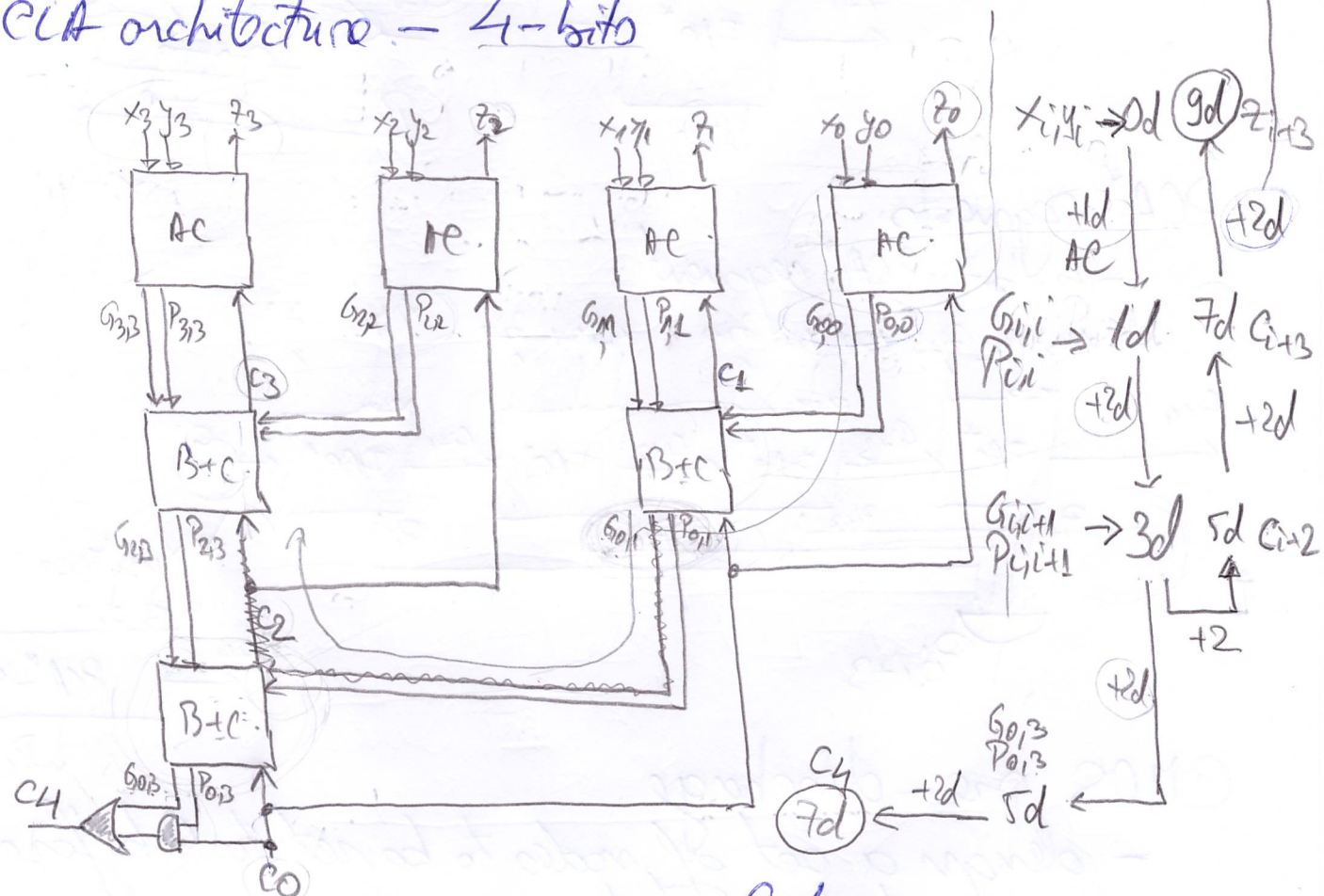
B+C cell



AC cell:



ML-CLA architecture – 4-bits



$D^{Z}_{ML-CLA-4} = 9d$ $\quad$ $D^{Cout}_{ML-CLA-4} = 7d$!!

n bits → $\lceil \log_2 n \rceil$ levels of (B+C) cells

$\lceil 3 \rceil = 3$

$\lceil 3.05 \rceil = 4$

In general:

$$D_{ML\text{-}CLA\text{-}n} = 1d + 2(\lceil \log_2 n \rceil - 1)d + 2\lceil \log_2 n \rceil d + 2d$$

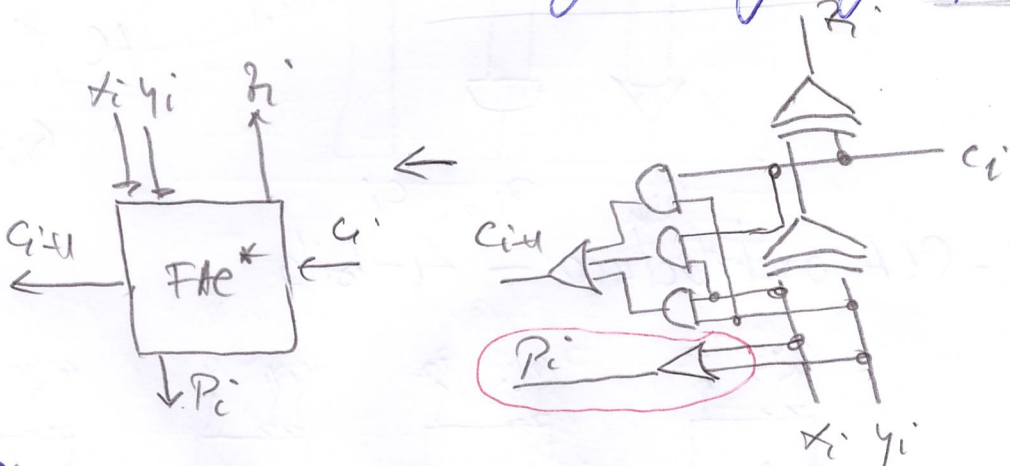
$$= (4\lceil \log_2 n \rceil + 1)d.$$

$$D^{Cout}_{ML\text{-}CLA\text{-}n} = (2\lceil \log_2 n \rceil + 3)d.$$

1.4.2. Carry Skip Adder. (CSkA).

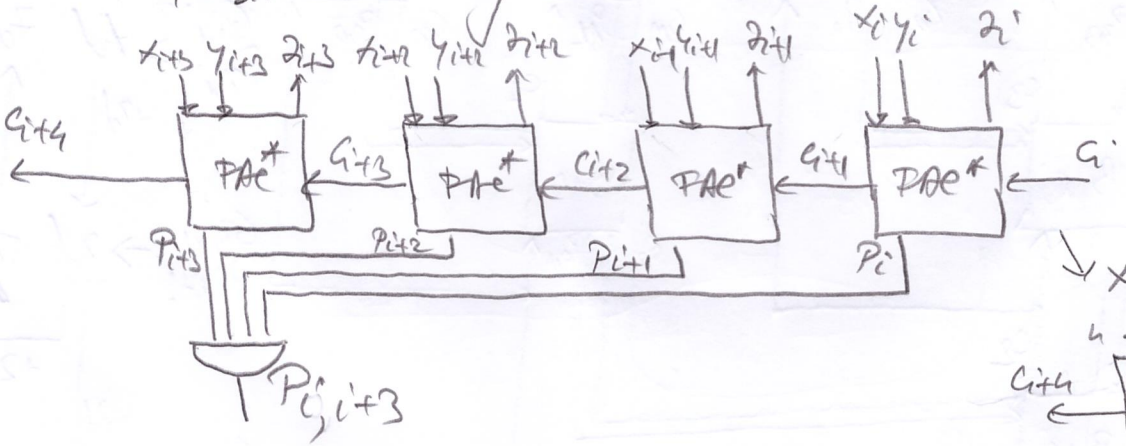
$c_{k+1} = G_{i,4} + P_{i,4} \cdot c_i$ — $P_{i,4}$ – simpler to obtain.

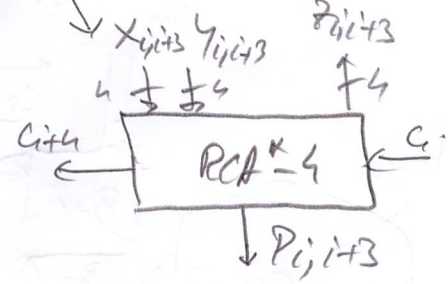
→ PAE generating signal P_i



RCA* segments.

– 4 bit RCA* segment:





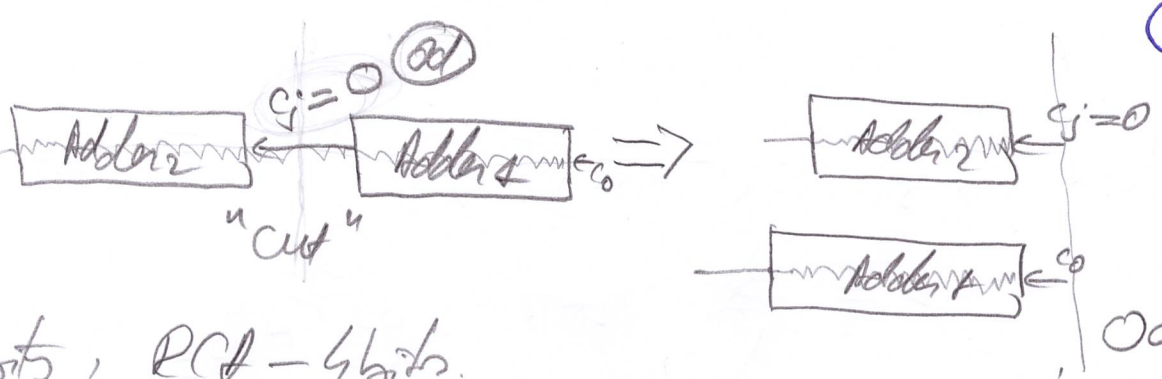
CMOS pre-discharge.

– design a set of nodes to be set to 0 before computation starts.

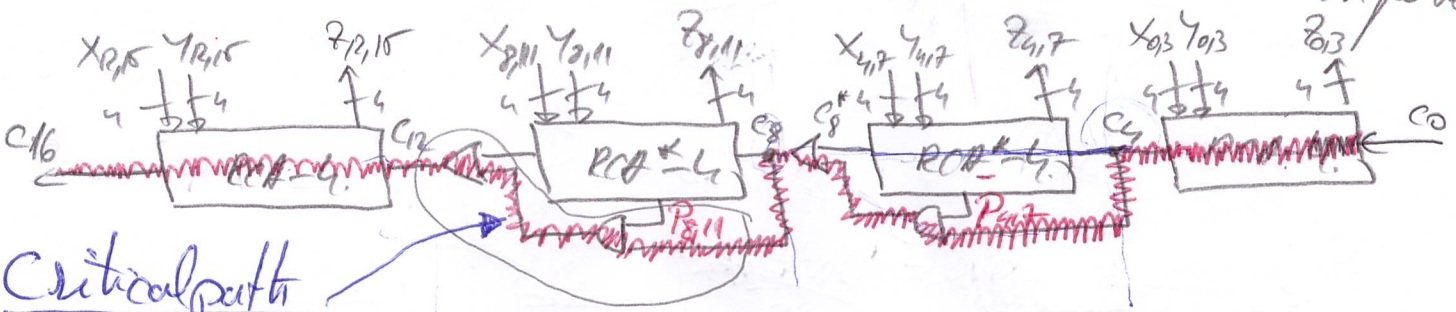
CSkA: all carries → pre-discharge.

⇒ if the final value of a carry is 0

→ that carry is correctly computed from 0d.



CSkA-16bits, RCA-4bits.



Critical path

Skip logic

Q: why use skip logic?

A: C_4, C_8, C_{12} = must be 1

↳ all carries at last the most significant carries must be 1

$C_8 = C_8^* + P_{4,7} \cdot C_4$

if $C_4 = 1$ ~~$C_4 = 0$~~

$P_{4,7} \cdot C_4 = 0 \rightarrow P_{4,7} = 0$

$P_{4,7} = P_4 \cdot P_5 \cdot P_6 \cdot P_7 = 0$

(7) P_i, $i \in [4,7]$

$P_i = 0$. $X_i + Y_i = 0$.

$\Rightarrow X_i = Y_i = 0$.

$\Rightarrow C_{i+1} = 0$.

$D^t_{CSkA-16} = \underbrace{8d}_{C_4} + \underbrace{2d}_{C_8} + \underbrace{2d}_{C_{12}} + \underbrace{8d}_{Z_{15}} = 20d$

$D^{Cout}_{CSkA-16} = \underbrace{8d}_{C_4} + \underbrace{2d}_{C_8} + \underbrace{2d}_{C_{12}} + \underbrace{8d}_{C_{16}} = 20d.$