

3.8

Carry Lookahead Adder

Consider addition of two n -bit numbers:

$$N_1 = b_{n-1} b_{n-2} \dots b_1 b_0 +$$

$$N_2 = \cancel{a}_{n-1} \cancel{a}_{n-2} \dots \cancel{a}_1 \cancel{a}_0$$

A carry to the n th bit: C_n is said to be:

(i) generated from the $(n-1)$ bits

if they are both 1: $b_{n-1} \cancel{a}_{n-1} = 1$.

We write

$$G_{n-1} = b_{n-1} \cancel{a}_{n-1}$$

G for generator bit

(ii) propagated from the $(n-m)$ bits if all the in-between bits added up to 1s, and there was a carry into the $(n-m)$ bits:

from $(n-1)$ to n th bit.

there is a carry into the $(n-m)$ bit.

$$\begin{array}{r}
 b_{n-1} \quad b_{n-2} \quad \dots \quad b_{n-m} \quad b_{n-m-1} \quad \dots \quad b_1 \quad b_0 \\
 + \quad \textcircled{0}_{n-1} \quad \textcircled{0}_{n-2} \quad \dots \quad \textcircled{0}_{n-m} \quad \textcircled{0}_{n-m-1} \quad \dots \quad \textcircled{0}_1 \quad \textcircled{0}_0 \\
 \hline
 1 \quad 1 \quad \dots \quad 1
 \end{array}$$

Then all these added to 1 s there will be a carry here.

Two bits add-up to 1 if:

① $1 + 0 = 1$, or

② $0 + 1 = 1$, or

$$P_{n-i} = b_{n-i} \oplus \textcircled{0}_{n-i}$$

(p for propagator).

The propagated carry C_{n-m} propagator as given by:

$$C_n = \underbrace{P_{n-1} P_{n-2} P_{n-3} \dots P_{n-m}}_{\text{all 1s}} C_{n-m}$$

↑
carry going in.

for the carrier:

0th input bit carry = C_0

$$C_1 = G_0 + P_0 C_0$$

↑
0th bit addition generates it

0th carry propagator.

$$C_2 = G_1 + P_1 C_1$$

$$\Rightarrow C_2 = G_1 + P_1 (G_0 + P_0 C_0) \\ = G_1 + P_1 G_0 + P_1 P_0 C_0$$

$$C_3 = G_2 + P_2 C_2$$

$$= G_2 + P_2 (G_1 + P_1 G_0 + P_1 P_0 C_0)$$

$$= G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0$$

generated by b_2, a_2'

generated by b_1, a_1' and propagated by b_2, a_2'

generated by b_0, a_0' and propagated ...

These equations are implementing a carry lookahead ad for addition.

The sum bits:

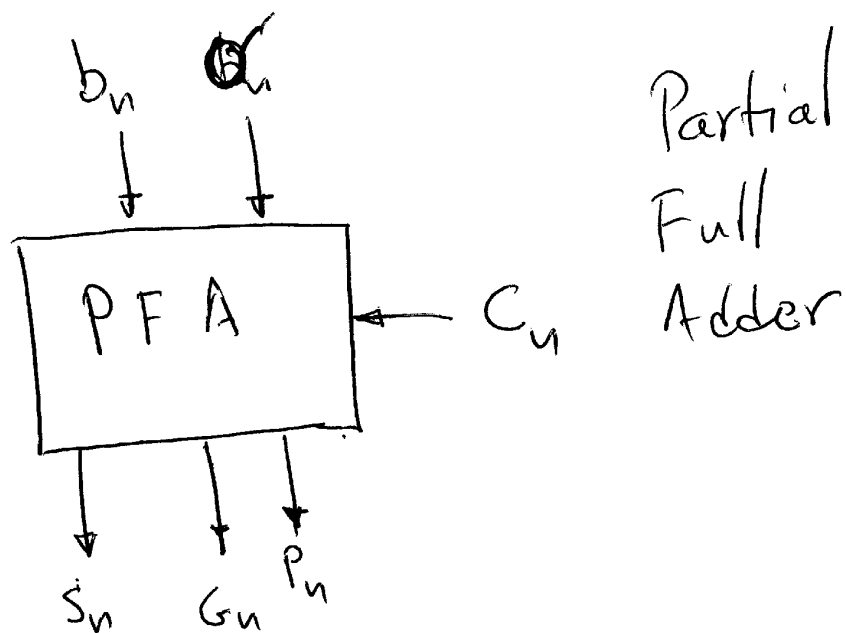
S_n are 1 if

- $b_n = 1, a_n = 0, c_n = 0$
- $b_n = 0, a_n = 1, c_n = 0$
- $b_n = 1, a_n = 1, c_n = 1$

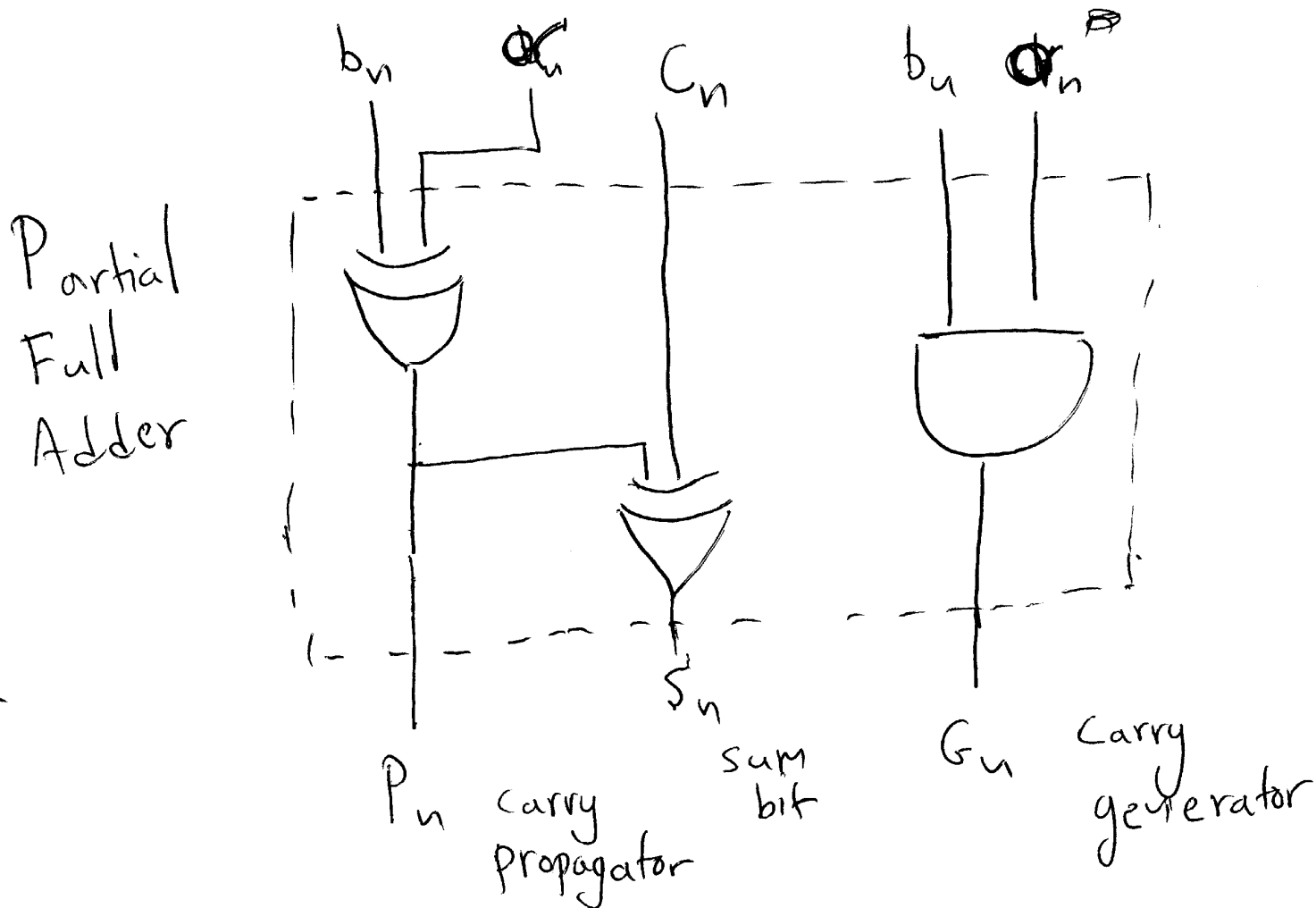
$$\Rightarrow S_n = (b_n \oplus a_n) \oplus c_n \\ = P_n \oplus c_n$$

1 if odd # of 1s

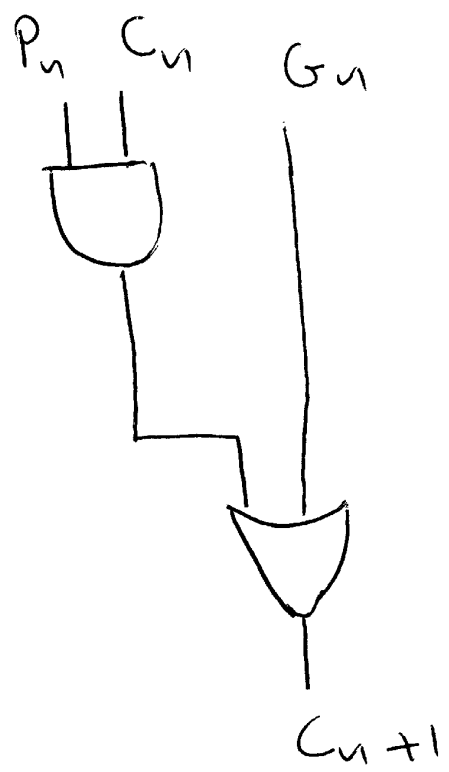
- Define:



- implemented by:

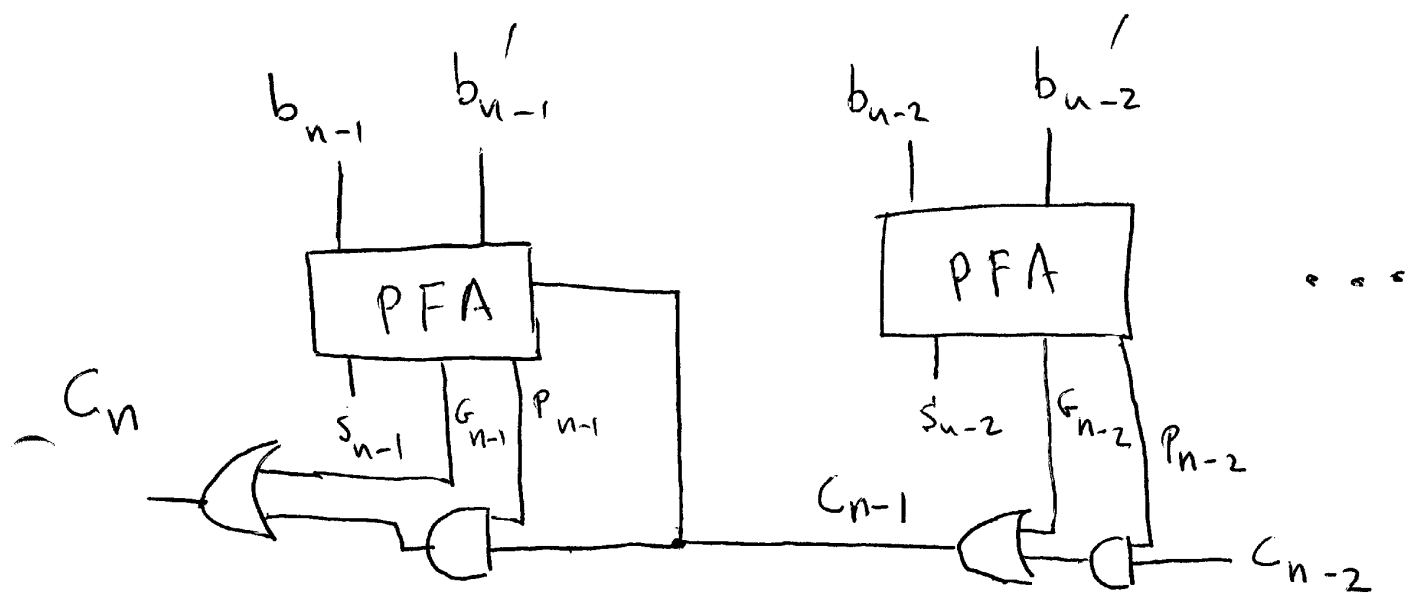


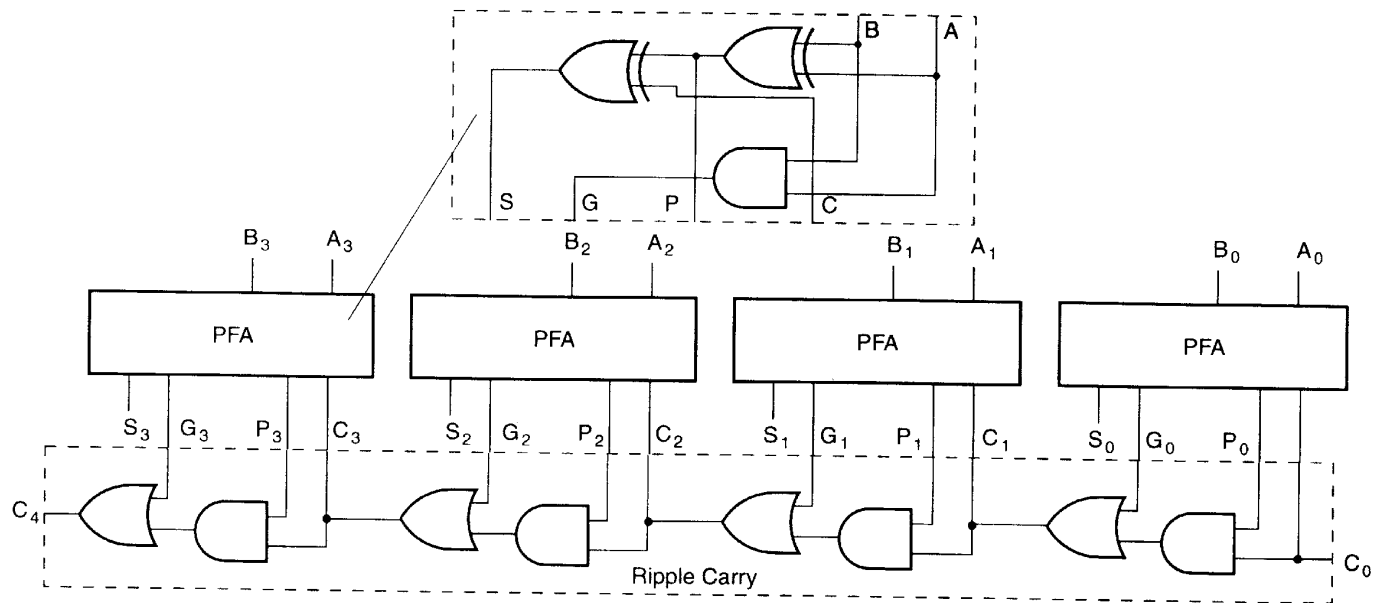
For the new carry out: C_{n+1}



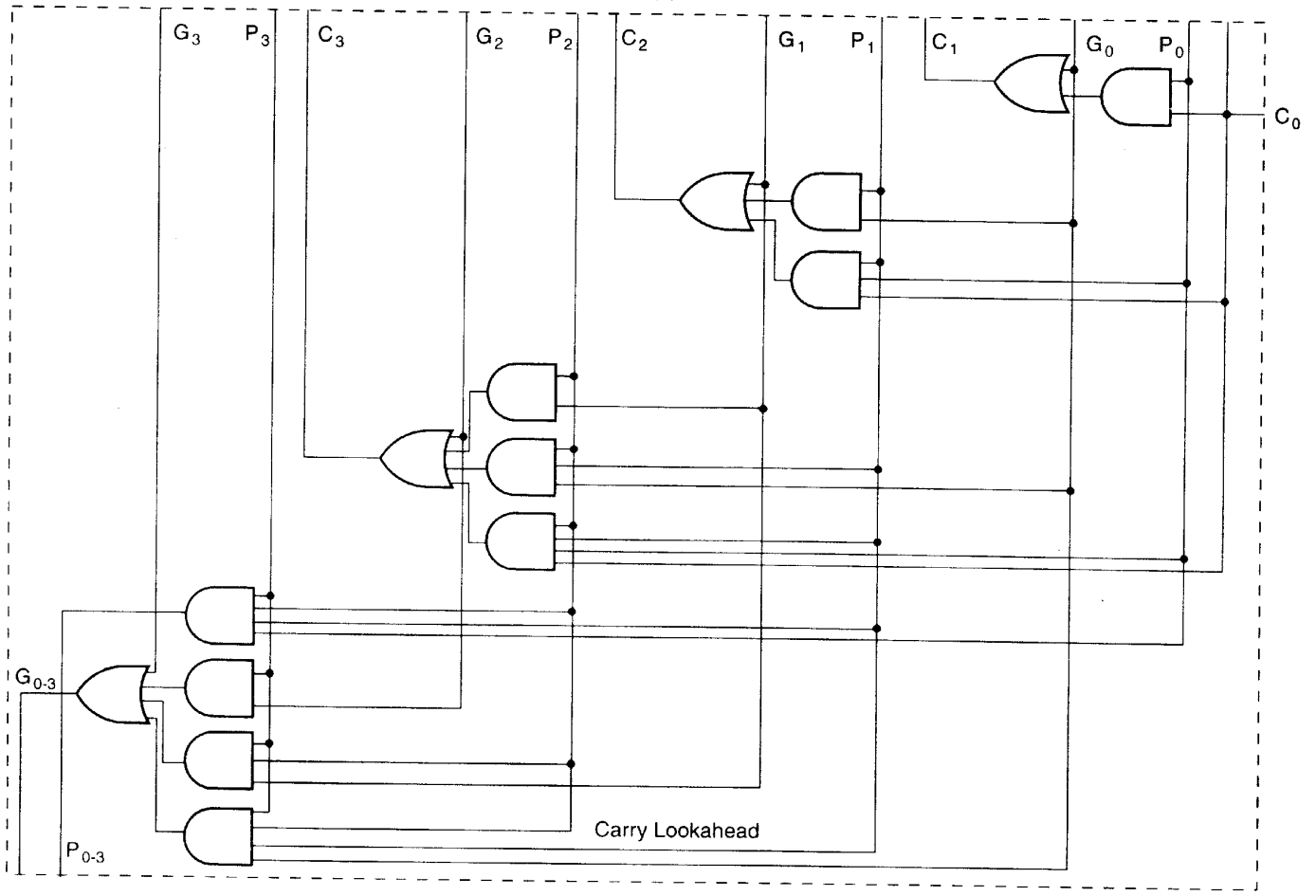
either generated or propagated.

The full-adder is:





(a)



(b)

FIGURE 3-29
Development of a Carry Lookahead Adder

Correction

6

- The longest/slowest path through the circuit is for the computation of S_3 (or s_1, s_2):

$$S_3 = P_3 \oplus C_3$$

where

$$C_3 = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0$$

Here is why:

stage 1: $P_0 = b_0 \oplus a_0, P_1 = b_1 \oplus a_1, \dots, P_3 = b_3 \oplus a_3$

- requires $2t_d$. (for \oplus)

stage 2

The carry logic requires $2t_d$:
(1 for ANDs + 1 for the OR)

stage 3

$$S_n = P_n \oplus C_n \quad \text{for } n=1, 2, 3.$$

require $2t_d$.

- Total = $6t_d$