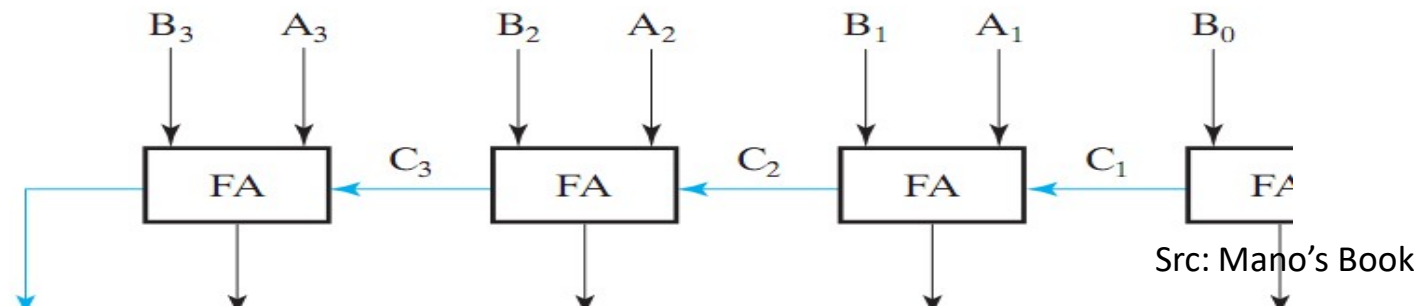


Carry Look Ahead Generator

Binary Parallel Adder (Again)

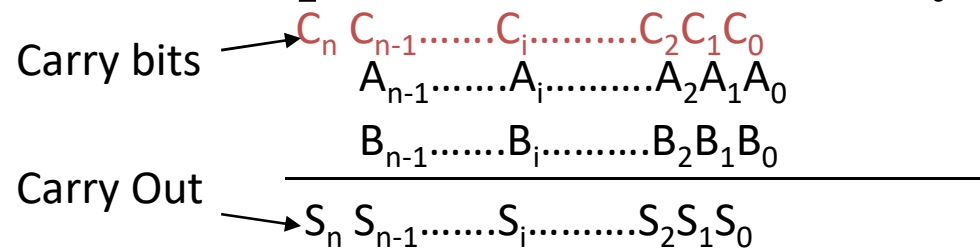
- To add n-bit numbers:
- Use n Full-Adders in parallel
- The carries propagate as in addition by hand



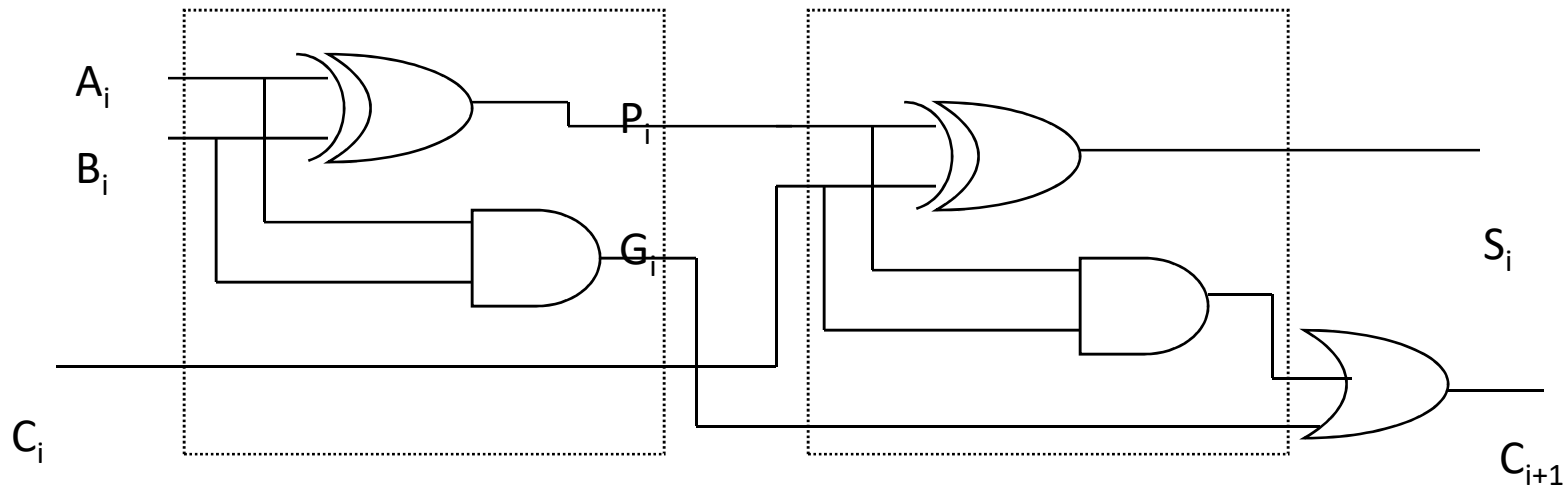
This adder is called *ripple carry adder*

Carry Look Ahead Addder

- How to reduce propagation delay of ripple carry adders?
- **Carry look ahead adder:** All carries are computed as a function of C_0 (independent of n !)
- It works on the following standard principles:
 - A carry bit is generated when both input bits A_i and B_i are 1, or
 - When one of input bits is 1, and a carry in bit exists



Carry Look Ahead Addder



The internal signals are given by:

$$P_i = A_i \oplus B_i$$

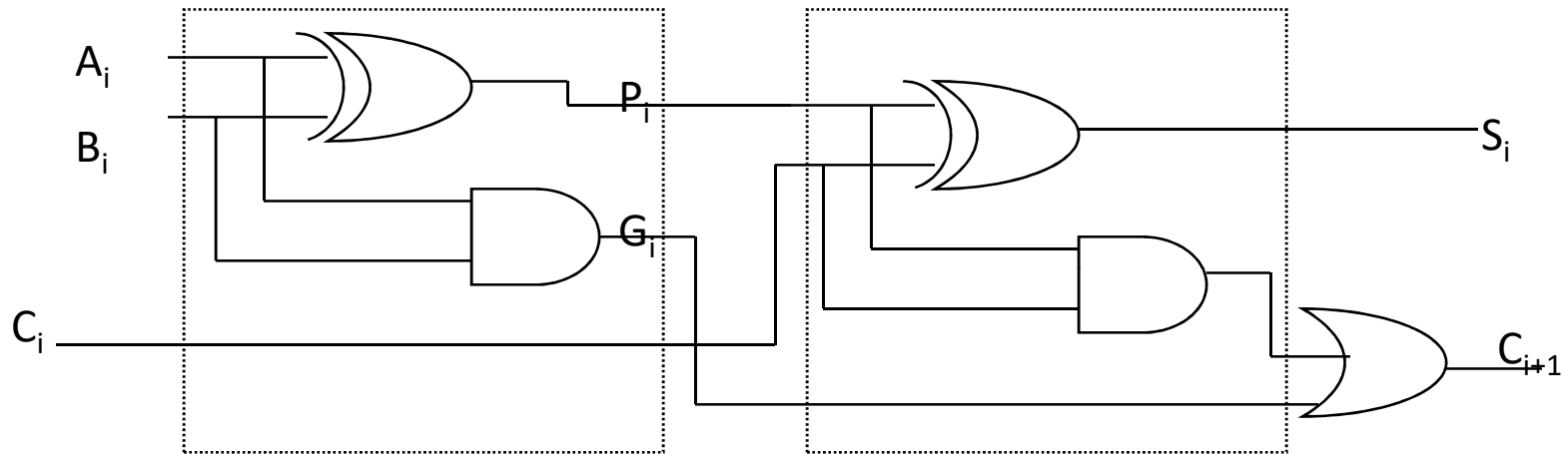
$$G_i = A_i \cdot B_i$$

Carry Generate G_i : $C_{i+1} = 1$ when $G_i = 1$, regardless of the input carry C_i

Carry Propagate P_i : Propagates C_i to C_{i+1}

Note: P_i and G_i depend only on A_i and B_i !

Carry Look Ahead Addder



The internal signals are given by:

$$P_i = A_i \oplus B_i$$

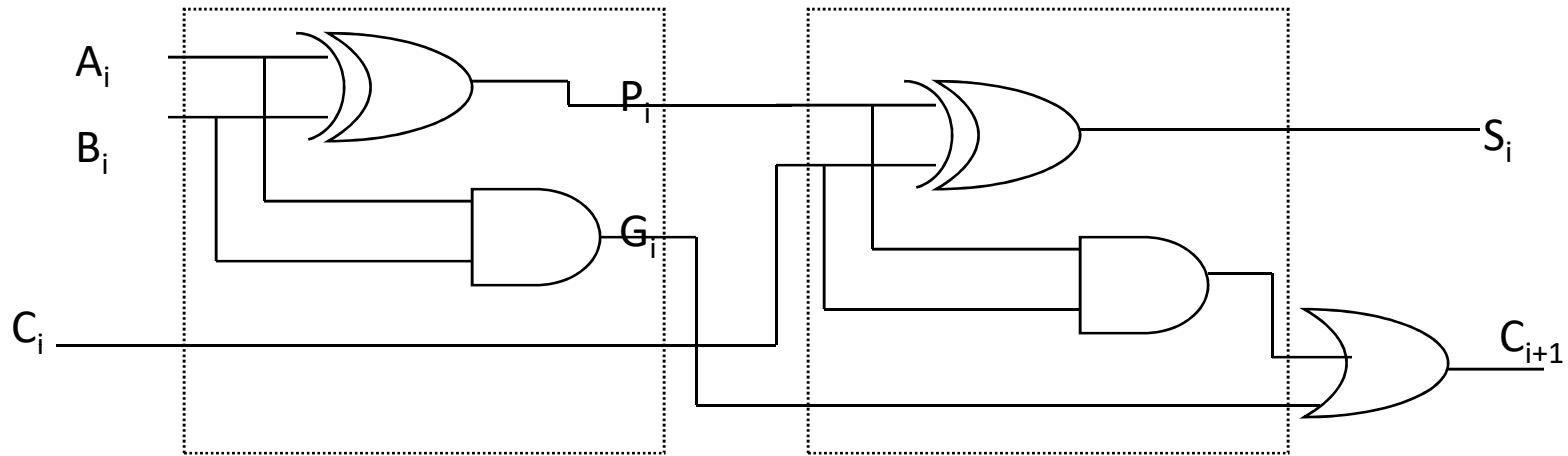
$$G_i = A_i \cdot B_i$$

The output signals are given by:

$$S_i = P_i \oplus C_i$$

$$C_{i+1} = G_i + P_i C_i$$

Carry Look Ahead Addder



The carry outputs for various stages can be written as:

$$C_1 = G_0 + P_0C_0$$

$$C_2 = G_1 + P_1C_1 = G_1 + P_1(G_0 + P_0C_0) = G_1 + P_1G_0 + P_1P_0C_0$$

$$C_3 = G_2 + P_2C_2 = G_2 + P_2G_1 + P_2P_1G_0 + P_2P_1P_0C_0$$

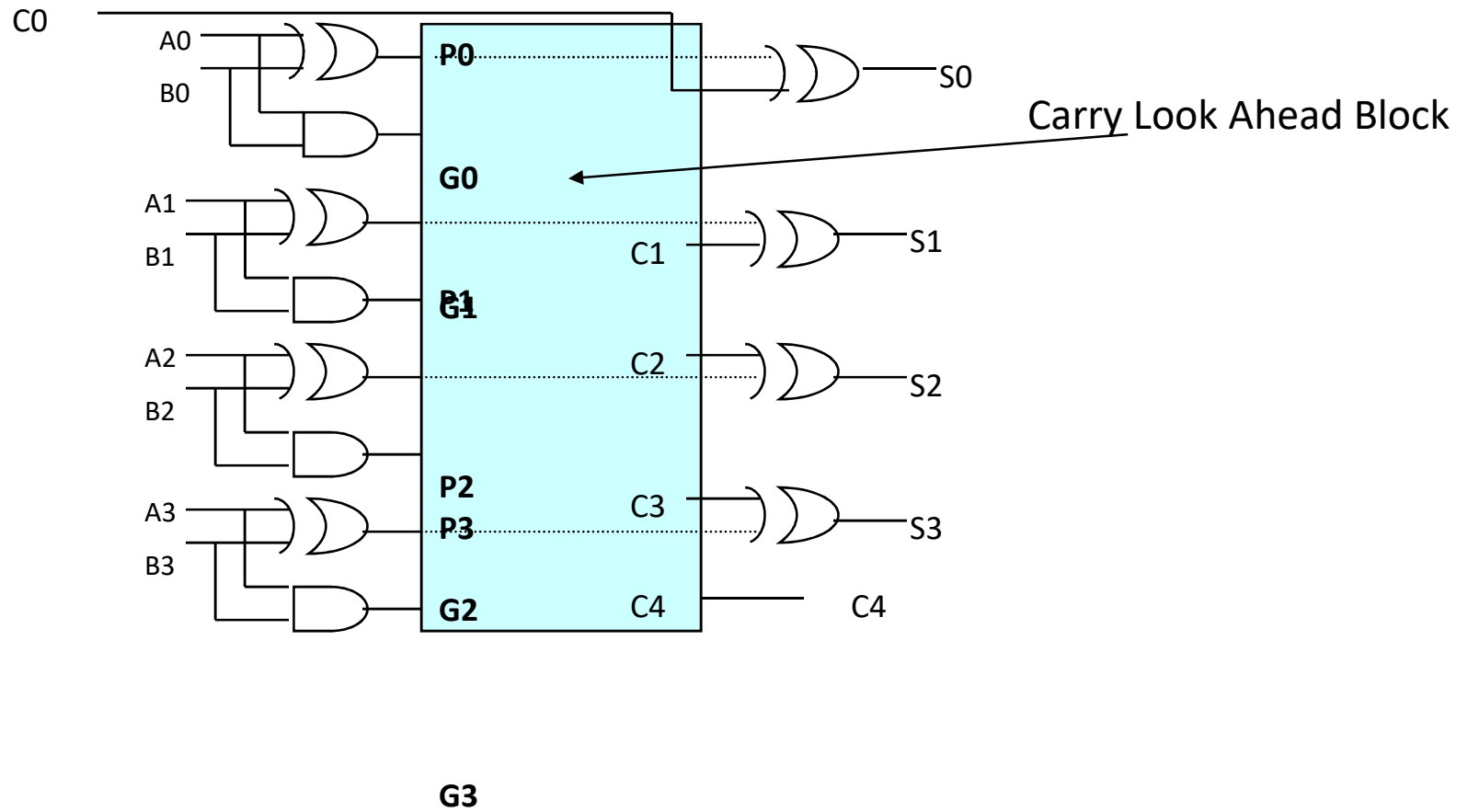
$$C_4 = G_3 + P_3C_3 = G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1G_0 + P_3P_2P_1P_0C_0$$

Carry Look Ahead Adder

Conclusion:

- Each carry bit can be expressed in terms of the input carry C_0 , and not based on its preceding carry bit
- Each carry bit can be expressed as a SOP, and can be implemented using a two-level circuit, i.e. a gate delay of $2T$

Carry Look Ahead Adder



Carry Look Ahead Adder

Steps of operation:

- All P and G signals are initially generated. Since both XOR and AND can be executed in parallel. Total delay = $1T$
- The Carry Look Ahead block will generate the four carry signals C_4, C_3, C_2, C_1 . Total delay = $2T$
- The four XOR gates will generate the Sums. Total delay = $1T$

Total delay before the output can be seen = $4T$

Cont...

Compared with the Ripple Adder delay of $9T$, this is an improvement of more than 100%

CLA adders are implemented as 4-bit modules, that can together be used for implementing larger circuits