- The ***carry propagation time*** is an important attribute of the adder because it limits the speed with which two numbers are added.
- To reduce the carry propagation delay time:
  1) Employ faster gates with reduced delays.
  2) Employ the principle of Carry Lookahead Logic.

Proof: ***(using carry lookahead logic)***

$$P_i = A_i \oplus B_i$$

$$G_i = A_i B_i$$

The output sum and carry are:

$$S_i = P_i \oplus C_i$$

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

- ✓ G_i-called a ***carry generate***, and it produces a carry of 1 when both A_i and B_i are 1.
- ✓ P_i-called a ***carry propagate***, it determines whether a carry into stage i will propagate into stage $i + 1$.
- ✓ The ***Boolean function*** for the carry outputs of each stage and substitute the value of each C_i form the previous equations:

$$\left\{\begin{array}{l} C_0 = input\ carry \\ C_1 = G_0 + P_0 C_0 \\ C_2 = G_1 + P_1 C_1 = G_1 + P_1(G_0 + P_0 C_0) \\ \quad = G_1 + P_1 G_0 + P_1 P_0 C_0 \\ C_3 = G_2 + P_2 C_2 = G_2 + P_2(G_1 + P_1 G_0 + P_1 P_0 C_0) \\ \quad = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0 \end{array}\right\}$$

- The three Boolean functions C_1, C_2 and C_3 are implemented in the ***carry lookahead generator.***

> ***The two level-circuit for the output carry C_4 is not shown, it can be easily derived by the equation.***

- C_3 does not have to wait for C_2 and C_1 to propagate, in fact C_3 is propagated at the same time as C_1 and C_2.