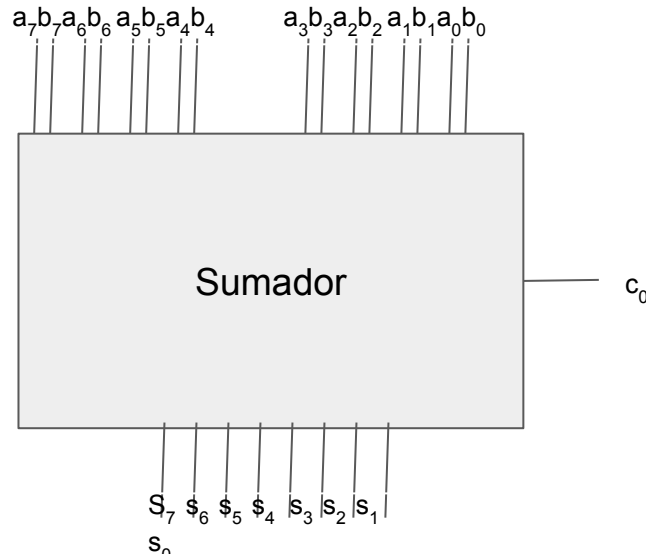


Proyecto 2025

Sumador Lookahead tree adder 8 bits

Lookahead tree adder

- El cálculo del propagar y generar permiten realizar sumas parciales.
- Acelera cálculo del acarreo y lo calcula en paralelo usando p y g.



Lookahead tree adder - 8 bits

Se calcula el valor de propagado y generado para el cálculo de sumas parciales.

$$p_i = a_i + b_i$$

$$g_i = a_i * b_i$$

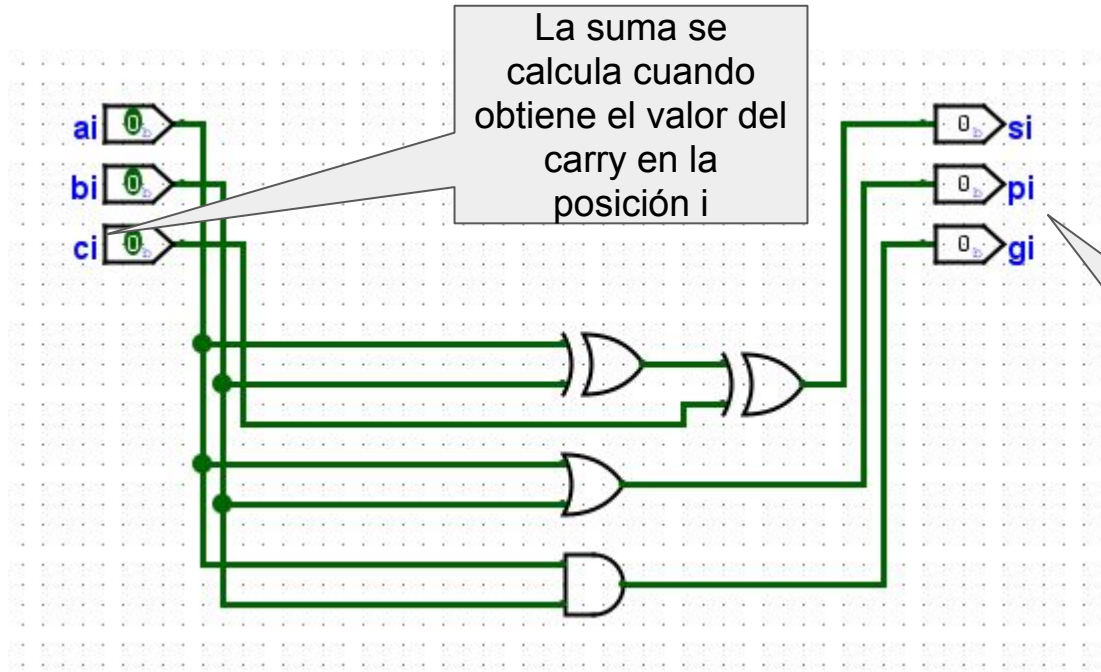
$$s_i = a_i \text{ xor } b_i \text{ xor } c_i$$

Repaso

Teniendo en cuenta dos bits, ¿cuándo generan y cuándo propagan?

a	b	generan?	propagan?
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

Bloque de nivel 0 - FA



$$p_i = a_i + b_i$$

$$g_i = a_i * b_i$$

$$s_i = a_i \text{ xor } b_i \text{ xor } c_i$$

Los valores p y g se calculan en el momento que se tienen los operandos a y b.

Es un Full adder que además calcula p y g.

Generalizando

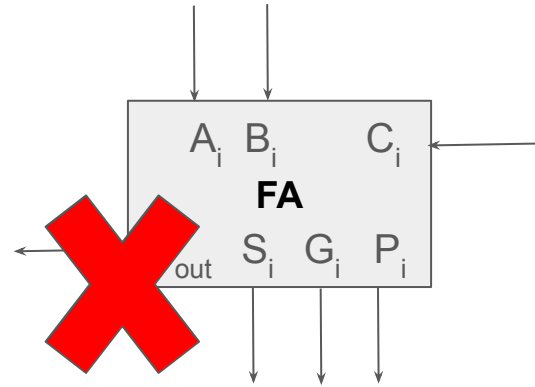
Tendremos 8 bloques tipo Full Adder, a los cuales ingresan un par de bits de los números A y B para una posición i, además ingresa un carry para esa posición.

Input:

A_i , B_i , C_i

Output:

C_{out} , S_i , G_i y P_i



Bloques de nivel superior

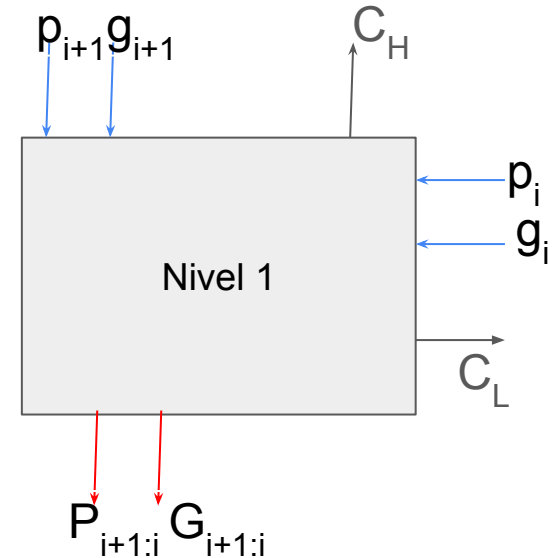
A medida que avanzamos en el árbol de acarreo, los valores p y g se combinan de manera jerárquica para generar los valores de acarreo para bloques más grandes (pares de bits, cuartetos de bits, etc.). Para cada nivel superior, los valores p y g se combinan para calcular los valores correspondientes de grupos de bits.

Nivel 1 - bloques agrupados de 2 bits

En este caso para dos bits se calcula p y g del siguiente modo:

$$P_{i+1:i} = p_{i+1} * p_i$$

$$G_{i+1:i} = g_{i+1} + (p_{i+1} * g_i)$$



Nivel 1 - bloques agrupados de 2 bits

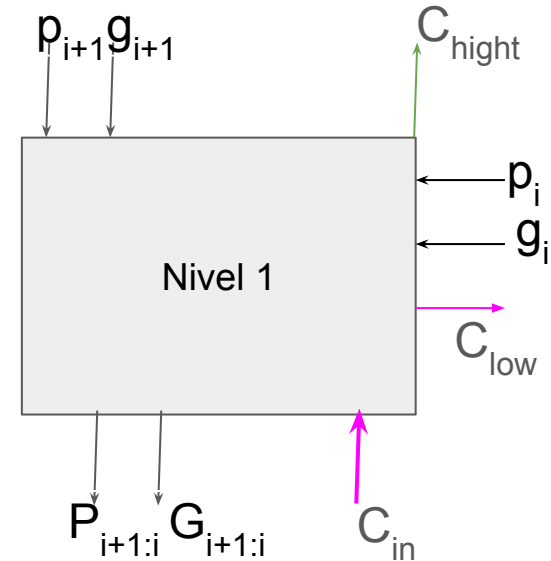
En este caso para dos bits se calcula p y g del siguiente modo:

$$P_{i+1:i} = p_{i+1} * p_i$$

$$G_{i+1:i} = g_{i+1} + (p_{i+1} * g_i)$$

$$C_{\text{hight}} = G_{i+1:i} + P_{i+1:i} * C_{\text{in}}$$

$$C_{\text{low}} = C_{\text{in}}$$



Nivel 1 - bloques agrupados de 2 bits

En este caso para dos bits se calcula p y g del siguiente modo:

$$P_{i+1:i} = p_{i+1} * p_i$$

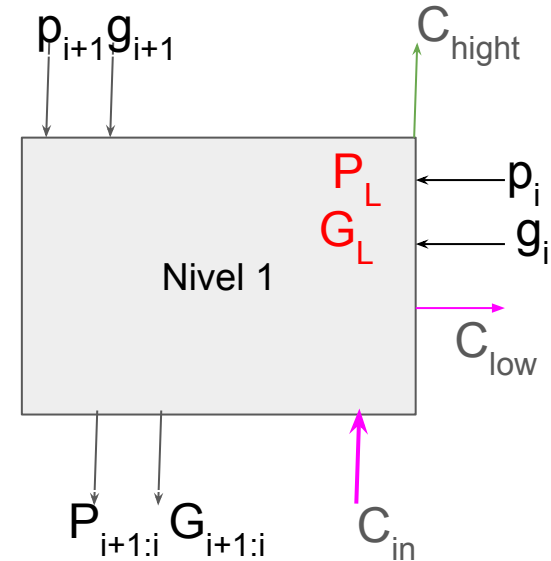
$$G_{i+1:i} = g_{i+1} + (p_{i+1} * g_i)$$

$$C_{\text{high}} = G_L + P_L * C_{\text{in}}$$

$$C_{\text{low}} = C_{\text{in}}$$

Usando C_{in} del bit menos significativo puedo generar los acarrees para los otros bloques.

El acarreo para el bloque de arriba, se calcula usando el acarreo que ingresa de la parte baja (C_{in}) junto con los P y G de la parte baja del bloque.



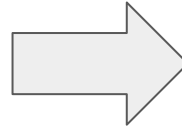
Generalizando

A partir de los valores generados y propagados que recibe un bloque, se podrían calcular los generados y propagados del siguiente nivel.

$$\left. \begin{aligned} P_{i+1:i} &= p_{i+1} * p_i \\ G_{i+1:i} &= g_{i+1} + (p_{i+1} * g_i) \end{aligned} \right\}$$

Si llamamos a:

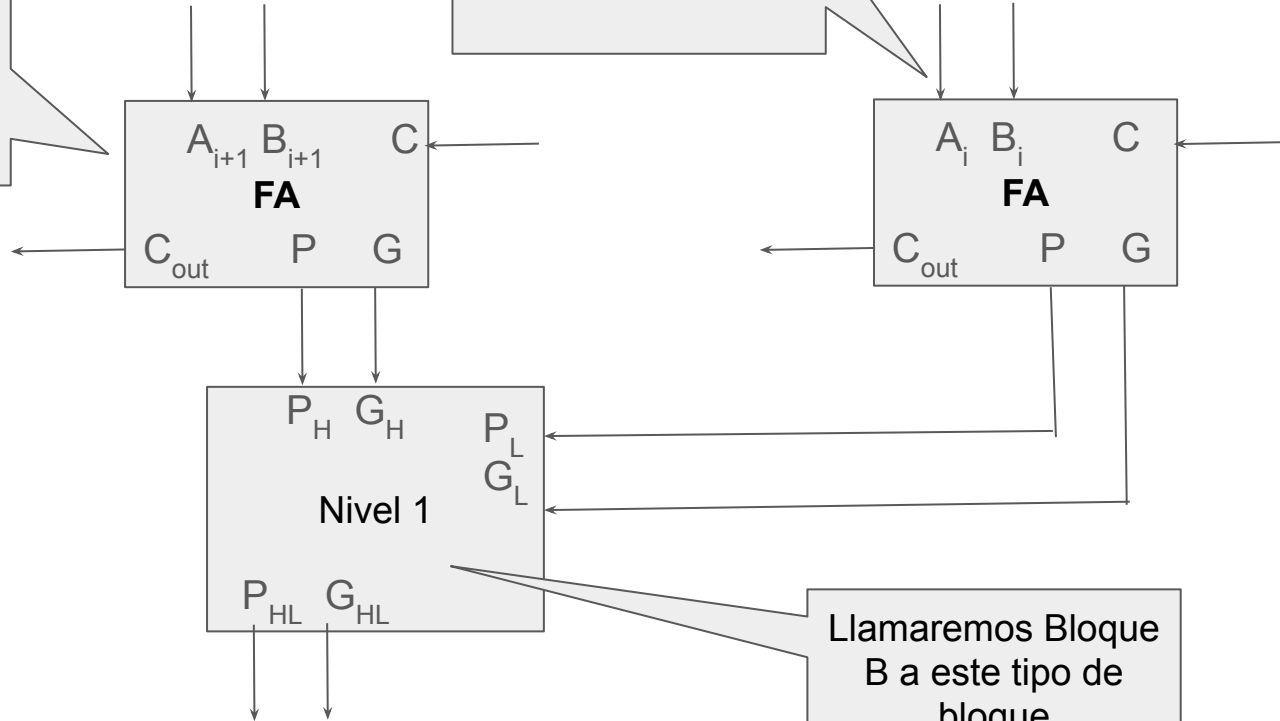
- p_i como P_{Low}
- g_i como G_{Low}
- p_{i+1} como P_{High}
- g_{i+1} como G_{High}

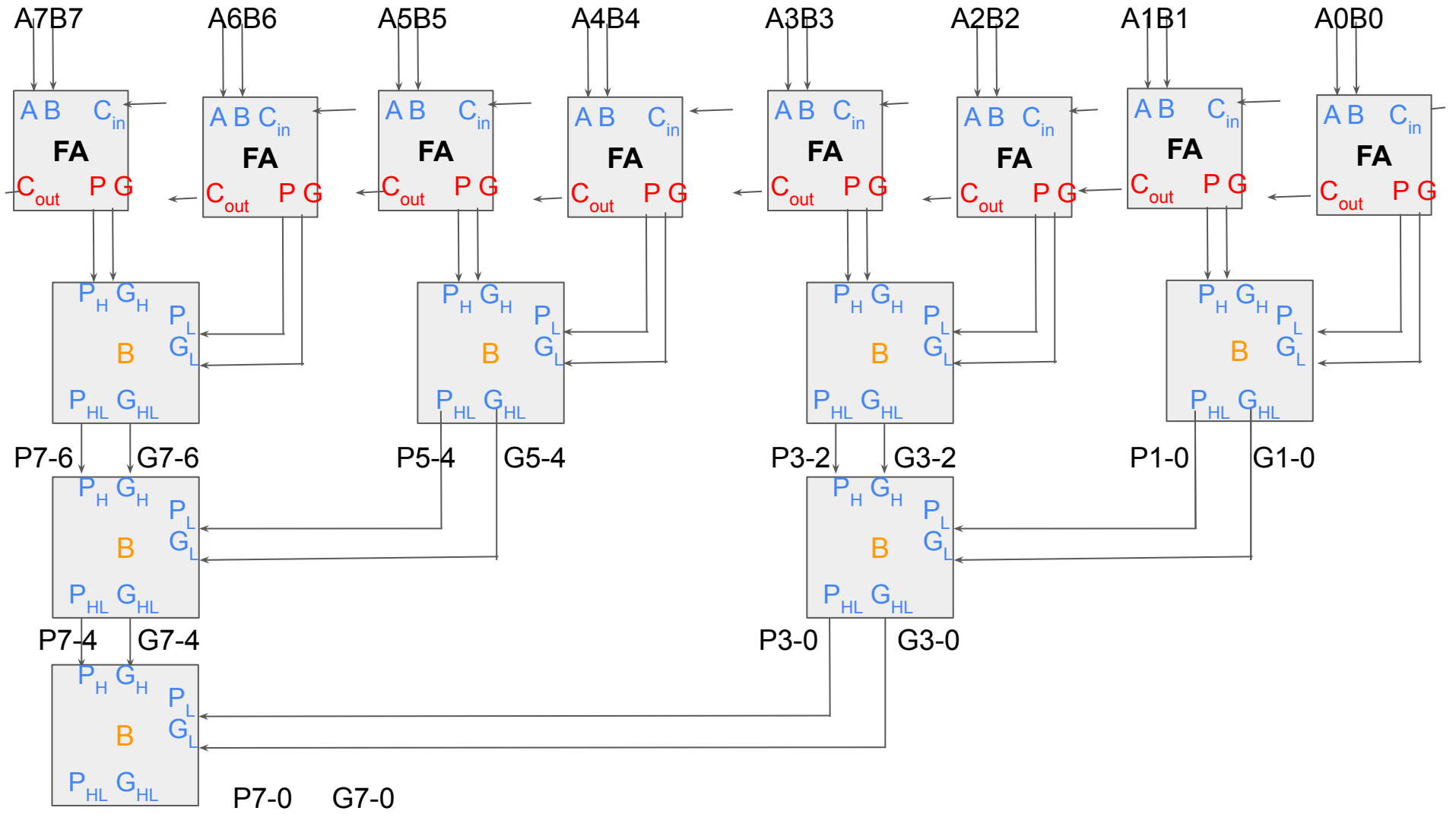


$$\begin{aligned} P_{\text{HL}} &= P_{\text{High}} * P_{\text{Low}} \\ G_{\text{HL}} &= G_{\text{High}} + (P_{\text{High}} * G_{\text{Low}}) \end{aligned}$$

Siendo el Full Adder en una posición genérica $i+1$ el bit **High**

Siendo el Full Adder en una posición genérica i el bit **Low**





Cálculo de carry

Cada bloque B calcula dos carries para dos bloques B de niveles anteriores, a los que llamaremos Carry Low y Carry High.

$$C_L = C_{in}$$

$$C_H = G_L + P_L * C_{in}$$

El valor de C_{in} que recibe el bloque B es pasado directamente como C_L .
El valor de CH se calcula a partir de los valores G_L , P_L y C_{in} que recibió el bloque.

Siendo el Full Adder en una posición genérica $i+1$ el bit **High**

Siendo el Full Adder en una posición genérica i el bit **Low**

